aws

Guida per l'utente

AWS AppConfig

AWS AppConfig: Guida per l'utente

Table of Contents

Cos'è AWS AppConfig?

AWS AppConfigi flag di funzionalità e le configurazioni dinamiche aiutano i produttori di software a regolare in modo rapido e sicuro il comportamento delle applicazioni negli ambienti di produzione senza implementare codice completo. AWS AppConfigaccelera la frequenza di rilascio del software, migliora la resilienza delle applicazioni e aiuta a risolvere più rapidamente i problemi emergenti. Con i flag di funzionalità, è possibile rilasciare gradualmente nuove funzionalità agli utenti e misurare l'impatto di tali modifiche prima di distribuire completamente le nuove funzionalità a tutti gli utenti. Grazie ai flag operativi e alle configurazioni dinamiche, è possibile aggiornare elenchi di blocchi, elenchi di autorizzazioni, limiti di limitazione, livello di dettaglio della registrazione ed eseguire altre ottimizzazioni operative per rispondere rapidamente ai problemi negli ambienti di produzione.

> **Note**
>
> AWS AppConfigè una funzionalità diAWS Systems Manager.

Migliora l'efficienza e rilascia le modifiche più velocemente

L'utilizzo di feature flag con nuove funzionalità accelera il processo di rilascio delle modifiche agli ambienti di produzione. Invece di affidarsi a rami di sviluppo di lunga durata che richiedono fusioni complicate prima di un rilascio, i feature flag consentono di scrivere software utilizzando lo sviluppo basato su trunk. I flag di funzionalità consentono di implementare in modo sicuro il codice di versione preliminare in una pipeline CI/CD nascosta agli utenti. Quando siete pronti a rilasciare le modifiche, potete aggiornare il flag della funzionalità senza distribuire nuovo codice. Una volta completato il lancio, il flag può ancora funzionare come interruttore a blocchi per disabilitare una nuova funzionalità o funzionalità senza la necessità di ripristinare la distribuzione del codice.

Evita modifiche o guasti non intenzionali con le funzionalità di sicurezza integrate

AWS AppConfigoffre le seguenti funzionalità di sicurezza per evitare l'attivazione dei flag di funzionalità o l'aggiornamento dei dati di configurazione che potrebbero causare errori delle applicazioni.

- Validatori: un validatore garantisce che i dati di configurazione siano corretti sintatticamente e semanticamente prima di implementare le modifiche agli ambienti di produzione.
- Strategie di implementazione: una strategia di implementazione consente di rilasciare lentamente le modifiche agli ambienti di produzione nell'arco di minuti o ore.

- Monitoraggio e rollback automatico: AWS AppConfig si integra con Amazon CloudWatch per monitorare le modifiche alle applicazioni. Se l'applicazione non funziona correttamente a causa di una modifica errata della configurazione e tale modifica fa scattare un allarme CloudWatch, ripristina AWS AppConfig automaticamente la modifica per ridurre al minimo l'impatto sugli utenti dell'applicazione.

Implementazioni di feature flag sicure e scalabili

AWS AppConfigsi integra con AWS Identity and Access Management (IAM) per fornire un accesso preciso e basato sui ruoli al servizio. AWS AppConfigsi integra anche con AWS Key Management Service (AWS KMS) per la crittografia e il controllo. AWS CloudTrail Prima di essere rilasciati ai clienti esterni, tutti i controlli di AWS AppConfig sicurezza sono stati inizialmente sviluppati e convalidati da clienti interni che utilizzano il servizio su larga scala.

Casi d'uso di AWS AppConfig

Nonostante il contenuto della configurazione dell'applicazione possa variare notevolmente da un'applicazione all'altra, AWS AppConfig supporta i seguenti casi d'uso, che coprono un ampio spettro di esigenze dei clienti:

- Funziona con bandiere e interruttori: rilascia nuove funzionalità in modo sicuro ai tuoi clienti in un ambiente controllato. Ripristina istantaneamente le modifiche in caso di problemi.
- Ottimizzazione delle applicazioni: introduci con attenzione le modifiche alle applicazioni testando al contempo l'impatto di tali modifiche sugli utenti negli ambienti di produzione.
- Elenco consentito o elenco bloccato: controlla l'accesso alle funzionalità premium o blocca istantaneamente utenti specifici senza distribuire nuovo codice.
- Storage di configurazione centralizzato: mantieni i dati di configurazione organizzati e coerenti su tutti i carichi di lavoro. È possibile utilizzare AWS AppConfig per distribuire i dati di configurazione archiviati nell'archivio di configurazione AWS AppConfig ospitatoAWS Secrets Manager, Systems Manager Parameter Store o Amazon S3.

Vantaggi dell'utilizzo di AWS AppConfig

AWS AppConfigoffre i seguenti vantaggi per la tua organizzazione:

- Riduci i tempi di inattività imprevisti per i tuoi clienti

AWS AppConfig riduce i tempi di inattività dell'applicazione consentendo di creare regole per convalidare la configurazione. Le configurazioni non valide non possono essere implementate. AWS AppConfigoffre le seguenti due opzioni per la convalida delle configurazioni:

- Per la convalida sintattica, è possibile utilizzare lo schema JSON. AWS AppConfig convalida la configurazione utilizzando lo schema JSON per assicurarsi che le modifiche alla configurazione rispettino i requisiti dell'applicazione.
- Per la convalida semantica, AWS AppConfig puoi chiamare una AWS Lambda funzione di tua proprietà per convalidare i dati all'interno della configurazione.

- Implementa rapidamente le modifiche su una serie di obiettivi

  AWS AppConfigsemplifica l'amministrazione delle applicazioni su larga scala implementando le modifiche alla configurazione da una posizione centrale. AWS AppConfigsupporta le configurazioni archiviate nell'archivio di configurazione AWS AppConfig ospitato, nel Systems Manager Parameter Store, nei documenti Systems Manager (SSM) e Amazon S3. È possibile utilizzare AWS AppConfig con applicazioni ospitate su istanze EC2, AWS Lambda, container, applicazioni per dispositivi mobili o dispositivi IoT.

  Le destinazioni non devono essere configurate con l'agente SSM di Systems Manager o il profilo di istanza IAM richiesto da altre funzionalità di Systems Manager. Ciò significa che AWS AppConfig funziona con le istanze non gestite.
- Aggiornare le applicazioni senza interruzioni

  AWS AppConfig distribuisce le modifiche di configurazione ai target in fase di runtime senza un processo di compilazione pesante o senza eliminare i target dal servizio.
- Controllo della distribuzione delle modifiche nell'applicazione

  Quando si implementano modifiche alla configurazione degli obiettivi, AWS AppConfig consente di ridurre al minimo i rischi utilizzando una strategia di implementazione. Le strategie di implementazione consentono di implementare lentamente le modifiche alla configurazione del parco macchine. Se riscontri un problema durante l'implementazione, puoi ripristinare la modifica alla configurazione prima che raggiunga la maggior parte dei tuoi host.

Funzionamento di AWS AppConfig

Questa sezione fornisce una descrizione di alto livello di come AWS AppConfig funziona e di come iniziare.

1. Identifica i valori di configurazione nel codice che desideri gestire nel cloud

   Prima di iniziare a creare AWS AppConfig artefatti, ti consigliamo di identificare nel codice i dati di configurazione che desideri gestire dinamicamente utilizzando. AWS AppConfig I buoni esempi includono le opzioni di attivazione o disattivazione delle funzionalità, gli elenchi consentiti e bloccati, la verbosità dei log, i limiti del servizio e le regole di limitazione, solo per citarne alcuni.

   Se i dati di configurazione esistono già nel cloud, puoi sfruttare le funzionalità di AWS AppConfig convalida, implementazione ed estensione per semplificare ulteriormente la gestione dei dati di configurazione.

2. Crea uno spazio dei nomi dell'applicazione

   Per creare uno spazio dei nomi, create un AWS AppConfig artefatto chiamato applicazione. Un'applicazione è semplicemente un costrutto organizzativo come una cartella.

3. Crea ambienti.

   Per ogni AWS AppConfig applicazione, si definiscono uno o più ambienti. Un ambiente è un raggruppamento logico di obiettivi, ad esempio applicazioni in un `Production` ambiente `Beta` OR, AWS Lambda funzioni o contenitori. È inoltre possibile definire ambienti per i sottocomponenti dell'applicazione, ad esempio `WebMobile`, e. `Back-end`

   Puoi configurare gli CloudWatch allarmi Amazon per ogni ambiente. Il sistema monitora gli allarmi durante una distribuzione della configurazione. Se viene attivato un allarme, il sistema ripristina la configurazione.

4. Creazione di un profilo di configurazione

   Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di profilo. AWS AppConfigsupporta due tipi di profili di configurazione: flag di funzionalità e configurazioni a forma libera. I profili di configurazione Feature Flag archiviano i propri dati nell'archivio di configurazione AWS AppConfig ospitato e l'URI è semplice. `hosted` Per i profili di configurazione in formato libero, è possibile archiviare i dati nell'archivio di configurazione AWS AppConfig ospitato o in qualsiasi AWS servizio con cui si integraAWS AppConfig, come descritto in. Creazione di un profilo di configurazione in formato libero in AWS AppConfig

   Un profilo di configurazione può anche includere validatori facoltativi per garantire che i dati di configurazione siano sintatticamente e semanticamente corretti. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione torna ai dati di configurazione precedenti.

5. Distribuisci i dati di configurazione

   Quando si crea una nuova distribuzione, si specifica quanto segue:

   - Un ID dell'applicazione
   - Un ID del profilo di configurazione
   - Una versione di configurazione
   - Un ID di ambiente in cui si desidera distribuire i dati di configurazione
   - Un ID della strategia di implementazione che definisce la velocità con cui desiderate che le modifiche abbiano effetto

   Quando richiami l'azione StartDeploymentAPI, AWS AppConfig esegue le seguenti attività:

   1. Recupera i dati di configurazione dal data store sottostante utilizzando l'URI di posizione nel profilo di configurazione.
   2. Verifica che i dati di configurazione siano corretti dal punto di vista sintattico e semantico utilizzando i validatori specificati al momento della creazione del profilo di configurazione.
   3. Memorizza nella cache una copia dei dati in modo che sia pronta per essere recuperata dall'applicazione. Questa copia memorizzata nella cache è denominata dati distribuiti.

6. Recupera la configurazione

   È possibile configurare AWS AppConfig l'agente come host locale e fare in modo che l'agente effettui il polling AWS AppConfig per gli aggiornamenti della configurazione. L'agente richiama le azioni StartConfigurationSessione GetLatestConfigurationAPI e memorizza nella cache i dati di configurazione localmente. Per recuperare i dati, l'applicazione effettua una chiamata HTTP al server localhost. AWS AppConfig L'agente supporta diversi casi d'uso, come descritto inMetodi di recupero semplificati.

   Se AWS AppConfig Agent non è supportato per il tuo caso d'uso, puoi configurare l'applicazione AWS AppConfig per verificare la presenza di aggiornamenti di configurazione richiamando direttamente le azioni StartConfigurationSessione l'GetLatestConfigurationAPI.

Nozioni di base su AWS AppConfig

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di AWS AppConfig.

Visualizza altri AWS video sul YouTube canale Amazon Web Services.

I seguenti blog possono aiutarti a saperne di più su AWS AppConfig e sulle sue funzionalità:

- Utilizzo dei flag AWS AppConfig di funzionalità
- Migliori pratiche per la convalida dei flag di AWS AppConfig funzionalità e dei dati di configurazione

SDK

Per informazioni sugli SDK AWS AppConfig specifici della lingua, consulta le seguenti risorse:

- AWS Command Line Interface
- SDK AWS per .NET
- SDK AWS per C++
- SDK AWS per Go
- AWS SDK per Java V2
- AWSSDK per JavaScript
- SDK AWS per PHP V3
- AWS SDK for Python
- SDK AWS per Ruby V3

Prezzi di AWS AppConfig

Il prezzo AWS AppConfig si pay-as-you-go basa sui dati di configurazione e sul recupero dei flag di funzionalità. Si consiglia di utilizzare l'AWS AppConfigagente per ottimizzare i costi. Per ulteriori informazioni, consulta Prezzi di AWS Systems Manager.

Quote AWS AppConfig

Le informazioni sugli AWS AppConfig endpoint e sulle quote di servizio, insieme ad altre quote di Systems Manager, si trovano in. Riferimenti generali di Amazon Web Services

> **Note**
>
> Per informazioni sulle quote per i servizi che archiviano le configurazioni AWS AppConfig, consulta Informazioni sulle quote e limitazioni dell'archivio di configurazione.

Configurazione AWS AppConfig

Se non l'hai già fatto, registrati Account AWS e crea un utente amministrativo.

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente amministrativo e utilizza solo l'utente root per eseguire attività che richiedono l'accesso di un utente root.

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Creazione di un utente amministratore

Dopo la registrazione Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Creazione di un utente amministratore

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, assegna l'accesso amministrativo a un utente amministratore.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Concessione dell'accesso programmatico

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con l' AWS AWS Management Console esterno di. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro	Utilizza credenziali temporane e per firmare le richieste	Segui le istruzioni per l'interfa ccia che desideri utilizzare.

Quale utente necessita dell'accesso programmatico?	Per	Come
(Utenti gestiti nel centro identità IAM)	programmatiche agli AWS CLI AWS SDK o alle API. AWS	• Per la AWS CLI, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'utente.AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento agli AWS SDK e agli strumenti.
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Segui le istruzioni in Uso delle credenziali temporanee con AWS risorse nella Guida per l'utente IAM.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfaccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli AWS SDK e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimento agli SDK e agli AWS strumenti. • Per le AWS API, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

(Facoltativo) Configura le autorizzazioni per il rollback in base agli allarmi CloudWatch

Puoi configurare AWS AppConfig il ripristino a una versione precedente di una configurazione in risposta a uno o più CloudWatch allarmi Amazon. Quando configuri una distribuzione per rispondere agli CloudWatch allarmi, specifichi un ruolo AWS Identity and Access Management (IAM). AWS AppConfig richiede questo ruolo in modo da poter monitorare gli CloudWatch allarmi.

> **Note**
>
> Il ruolo IAM deve appartenere all'account corrente. Per impostazione predefinita, AWS AppConfig può monitorare solo gli allarmi di proprietà dell'account corrente. Se desideri configurare AWS AppConfig il rollback delle distribuzioni in risposta alle metriche di un

account diverso, devi configurare gli allarmi tra account. Per ulteriori informazioni, consulta la CloudWatch console interregionale per più account nella Amazon CloudWatch User Guide.

Utilizza le seguenti procedure per creare un ruolo IAM che AWS AppConfig consenta il rollback in base agli allarmi. CloudWatch Questa sezione include le seguenti procedure.

1. Passaggio 1: creare la politica di autorizzazione per il rollback in base agli allarmi CloudWatch
2. Passaggio 2: crea il ruolo IAM per il rollback in base CloudWatch agli allarmi
3. Fase 3: aggiunta di una relazione di trust

Passaggio 1: creare la politica di autorizzazione per il rollback in base agli allarmi CloudWatch

Utilizza la seguente procedura per creare una policy IAM che AWS AppConfig autorizzi a richiamare l'azione dell'`DescribeAlarms`API.

Per creare una politica di autorizzazione IAM per il rollback basata sugli allarmi CloudWatch

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON.
4. Sostituisci il contenuto predefinito nella scheda JSON con la seguente politica di autorizzazione, quindi scegli Avanti: Tag.

> **Note**
>
> Per restituire informazioni sugli allarmi CloudWatch compositi, all'operazione DescribeAlarmsAPI devono essere assegnate * le autorizzazioni, come mostrato qui. Non è possibile restituire informazioni sugli allarmi compositi se l'ambito `DescribeAlarms` è più ristretto.

```
{
        "Version": "2012-10-17",
        "Statement": [
```

```
            {
                "Effect": "Allow",
                "Action": [
                    "cloudwatch:DescribeAlarms"
                ],
                "Resource": "*"
            }
        ]
    }
```

5. Immettere i tag per questo ruolo, quindi scegliere Next: Review (Successivo: Revisione).
6. Nella pagina Revisione, inserisci il **`SSMCloudWatchAlarmDiscoveryPolicy`** campo Nome.
7. Scegli Crea policy. Il sistema visualizza di nuovo la pagina Policies (Policy).

Passaggio 2: crea il ruolo IAM per il rollback in base CloudWatch agli allarmi

Utilizza la procedura seguente per creare un ruolo IAM e assegnargli la policy creata nella procedura precedente.

Per creare un ruolo IAM per il rollback basato sugli allarmi CloudWatch

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, scegliere Roles (Ruoli) e quindi Create role (Crea ruolo).
3. In Select type of trusted entity (Seleziona tipo di entità attendibile), scegliere AWS service (Servizio).
4. Subito sotto Scegli il servizio che utilizzerà questo ruolo, scegli EC2: consente alle istanze EC2 di chiamare AWS i servizi per tuo conto, quindi scegli Avanti: Autorizzazioni.
5. Nella pagina Politica sulle autorizzazioni allegate, cerca SSM. CloudWatchAlarmDiscoveryPolicy
6. Scegliere questa policy, quindi selezionare Next: Tags (Successivo: tag).
7. Immettere i tag per questo ruolo, quindi scegliere Next: Review (Successivo: Revisione).
8. Nella pagina Crea ruolo, inserisci il **`SSMCloudWatchAlarmDiscoveryRole`** campo Nome ruolo, quindi scegli Crea ruolo.
9. Nella pagina Roles (Ruoli), scegliere il ruolo appena creato. Viene visualizzata la pagina Summary (Riepilogo).

Fase 3: aggiunta di una relazione di trust

Utilizza la procedura seguente per configurare il ruolo appena creato per considerare attendibile AWS AppConfig.

Per aggiungere una relazione di fiducia per AWS AppConfig

1. Nella pagina Summary (Riepilogo) per il ruolo creato in precedenza, scegliere la scheda Trust Relationships (Relazioni di trust), quindi scegliere Edit Trust Relationship (Modifica relazione di trust).
2. Modificare il criterio per includere solo "`appconfig.amazonaws.com`", come mostrato nell'esempio seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appconfig.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. Scegli Update Trust Policy (Aggiorna policy di trust).

Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig

Gli argomenti di questa sezione consentono di completare le seguenti attività in AWS AppConfig. Queste attività creano artefatti importanti per la distribuzione dei dati di configurazione.

1. Crea uno spazio dei nomi dell'applicazione

   Per creare uno spazio dei nomi di un'applicazione, si crea un elemento chiamato applicazione AWS AppConfig . Un'applicazione è semplicemente un costrutto organizzativo come una cartella.

2. Crea ambienti

   Per ogni AWS AppConfig applicazione, si definiscono uno o più ambienti. Un ambiente è un gruppo di AWS AppConfig destinazione di distribuzione logica, ad esempio le applicazioni in un `Production` ambiente `Beta` OR. È inoltre possibile definire ambienti per i sottocomponenti delle applicazioni, come`AWS Lambda functions`,`Containers`, `WebMobile`, e`Back-end`.

   Puoi configurare gli CloudWatch allarmi Amazon per ogni ambiente per ripristinare automaticamente le modifiche di configurazione problematiche. Il sistema monitora gli allarmi durante una distribuzione della configurazione. Se viene attivato un allarme, il sistema ripristina la configurazione.

3. Crea un profilo di configurazione

   Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di profilo. AWS AppConfig supporta due tipi di profili di configurazione: flag di funzionalità e configurazioni a forma libera. I profili di configurazione Feature Flag archiviano i propri dati nell'archivio di configurazione AWS AppConfig ospitato e l'URI è semplice. `hosted` Per i profili di configurazione in formato libero, è possibile archiviare i dati nell'archivio di configurazione AWS AppConfig ospitato o in un'altra funzionalità o AWS servizio di Systems Manager che si integra con AWS AppConfig, come descritto in. Creazione di un profilo di configurazione in formato libero in AWS AppConfig

   Un profilo di configurazione può anche includere validatori opzionali per garantire che i dati di configurazione siano corretti dal punto di vista sintattico e semantico. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione si interrompe prima di apportare modifiche ai target di configurazione.

> **Note**
>
> A meno che tu non abbia esigenze specifiche per l'archiviazione di segreti AWS Secrets Manager o la gestione dei dati in Amazon Simple Storage Service (Amazon S3), ti consigliamo di ospitare i dati di configurazione nell'archivio di configurazione ospitato in quanto offre AWS AppConfig la maggior parte delle funzionalità e dei miglioramenti.

Argomenti

Configurazioni di esempio

Utilizza AWS AppConfig, una funzionalità di AWS Systems Manager, per creare, gestire e distribuire rapidamente configurazioni di applicazioni. Una configurazione è una raccolta di impostazioni che influenzano il comportamento dell'applicazione. Ecco alcuni esempi.

Configurazione dei flag di funzionalità

La seguente configurazione dei flag di funzionalità abilita o disabilita i pagamenti mobili e i pagamenti predefiniti in base alla regione.

JSON

```
{
  "allow_mobile_payments": {
    "enabled": false
  },
  "default_payments_per_region": {
    "enabled": true
  }
}
```

YAML

```
---
allow_mobile_payments:
  enabled: false
default_payments_per_region:
  enabled: true
```

Configurazione operativa

La seguente configurazione in formato libero impone dei limiti al modo in cui un'applicazione elabora le richieste.

JSON

```
{
  "throttle-limits": {
    "enabled": "true",
    "throttles": [
      {
        "simultaneous_connections": 12
      },
      {
        "tps_maximum": 5000
      }
    ],
    "limit-background-tasks": [
      true
    ]
  }
}
```

YAML

```
---
throttle-limits:
  enabled: 'true'
  throttles:
  - simultaneous_connections: 12
  - tps_maximum: 5000
  limit-background-tasks:
```

```
  - true
```

Configurazione della lista di controllo degli accessi

La seguente configurazione a forma libera dell'elenco di controllo degli accessi specifica quali utenti o gruppi possono accedere a un'applicazione.

JSON

```
{
  "allow-list": {
    "enabled": "true",
    "cohorts": [
      {
        "internal_employees": true
      },
      {
        "beta_group": false
      },
      {
        "recent_new_customers": false
      },
      {
        "user_name": "Jane_Doe"
      },
      {
        "user_name": "John_Doe"
      }
    ]
  }
}
```

YAML

```
---
allow-list:
  enabled: 'true'
  cohorts:
  - internal_employees: true
  - beta_group: false
  - recent_new_customers: false
  - user_name: Jane_Doe
```

```
  - user_name: Ashok_Kumar
```

Informazioni sul profilo di configurazione (ruolo IAM)

Puoi creare il ruolo IAM che fornisce l'accesso ai dati di configurazione utilizzando AWS AppConfig. Oppure puoi creare tu stesso il ruolo IAM. Se crei il ruolo utilizzando AWS AppConfig, il sistema crea il ruolo e specifica una delle seguenti politiche di autorizzazione, a seconda del tipo di origine di configurazione scelta.

L'origine della configurazione è un segreto di Secrets Manager

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
             ],
            "Resource": [
                "arn:aws:secretsmanager:Regione AWS:account_ID:secret:secret_name-
a1b2c3"
            ]
        }
    ]
}
```

L'origine della configurazione è un parametro Parameter Store

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter"
            ],
            "Resource": [
                "arn:aws:ssm:Regione AWS:account_ID:parameter/parameter_name"
            ]
        }
```

```
    ]
    }
```

L'origine di configurazione è un documento SSM

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetDocument"
            ],
            "Resource": [
                "arn:aws:ssm:Regione AWS:account_ID:document/document_name"
            ]
        }
    ]
}
```

Se si crea il ruolo utilizzando AWS AppConfig, il sistema crea anche la seguente relazione di trust per il ruolo.

```
{

  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appconfig.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Creazione di uno spazio dei nomi per l'applicazione in AWS AppConfig

Le procedure descritte in questa sezione consentono di creare un elemento AWS AppConfig chiamato applicazione. Un'applicazione è semplicemente un costrutto organizzativo come una cartella che identifica lo spazio dei nomi dell'applicazione. Questo costrutto organizzativo ha una relazione con alcune unità di codice eseguibile. Ad esempio, è possibile creare un'applicazione chiamata MyMobileApp per organizzare e gestire i dati di configurazione per un'applicazione mobile installata dagli utenti. È necessario creare questi artefatti prima di poterli utilizzare AWS AppConfig per distribuire e recuperare i flag di funzionalità o i dati di configurazione in formato libero.

> **Note**
>
> È possibile utilizzarli AWS CloudFormation per creare AWS AppConfig artefatti, tra cui applicazioni, ambienti, profili di configurazione, distribuzioni, strategie di distribuzione e versioni di configurazione ospitate. Per ulteriori informazioni, consulta Riferimento al tipo di risorsa AWS AppConfig nella Guida dell'utente di AWS CloudFormation .

Argomenti

Creazione di un' AWS AppConfig applicazione (console)

Utilizzare la procedura seguente per creare un' AWS AppConfig applicazione utilizzando la AWS Systems Manager console.

Per creare un'applicazione

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Sulla scheda Applications (Applicazioni), scegliere Create application (Crea applicazione).
3. In Name (Nome), immettere un nome per l'applicazione.
4. In Description (Descrizione), immettere le informazioni sull'applicazione.

5. (Facoltativo) Nella sezione Estensioni, scegliete un'estensione dall'elenco. Per ulteriori informazioni, consulta Informazioni sulle AWS AppConfig estensioni.
6. (Facoltativo) Nella sezione Tag, inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
7. Scegli Crea applicazione.

AWS AppConfig crea l'applicazione e quindi visualizza la scheda Ambienti. Passa a Creazione di ambienti per l'applicazione in AWS AppConfig.

Creazione di un' AWS AppConfig applicazione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Tools for PowerShell creare un' AWS AppConfig applicazione.

Per creare un'applicazione passo dopo passo

1. Apri il AWS CLI.
2. Eseguite il comando seguente per creare un'applicazione.

   Linux

   ```
   aws appconfig create-application \
     --name A_name_for_the_application \
     --description A_description_of_the_application \
     --tags User_defined_key_value_pair_metadata_for_the_application
   ```

   Windows

   ```
   aws appconfig create-application ^
     --name A_name_for_the_application ^
     --description A_description_of_the_application ^
     --tags User_defined_key_value_pair_metadata_for_the_application
   ```

   PowerShell

   ```
   New-APPCApplication `
     -Name Name_for_the_application `
     -Description Description_of_the_application `
     -Tag Hashtable_type_user_defined_key_value_pair_metadata_for_the_application
   ```

Il sistema restituisce informazioni simili alle seguenti.

Linux

```
{
   "Id": "Application ID",
   "Name": "Application name",
   "Description": "Description of the application"
}
```

Windows

```
{
   "Id": "Application ID",
   "Name": "Application name",
   "Description": "Description of the application"
}
```

PowerShell

```
ContentLength    : Runtime of the command
Description      : Description of the application
HttpStatusCode   : HTTP Status of the runtime
Id               : Application ID
Name             : Application name
ResponseMetadata : Runtime Metadata
```

Creazione di ambienti per l'applicazione in AWS AppConfig

Per ogni AWS AppConfig applicazione, si definiscono uno o più ambienti. Un ambiente è un gruppo di AppConfig obiettivi di distribuzione logico, ad esempio applicazioni in un `Production` ambiente `Beta` OR, AWS Lambda funzioni o contenitori. È inoltre possibile definire ambienti per i sottocomponenti delle applicazioni, ad esempio `WebMobile`, e`Back-end`. Puoi configurare gli CloudWatch allarmi Amazon per ogni ambiente. Il sistema monitora gli allarmi durante una distribuzione della configurazione. Se viene attivato un allarme, il sistema ripristina la configurazione.

Prima di iniziare

Se desideri abilitare AWS AppConfig il ripristino di una configurazione in risposta a un CloudWatch allarme, devi configurare un ruolo AWS Identity and Access Management (IAM) con autorizzazioni per consentire AWS AppConfig la risposta agli allarmi. CloudWatch È possibile scegliere questo ruolo nella procedura seguente. Per ulteriori informazioni, consulta (Facoltativo) Configura le autorizzazioni per il rollback in base agli allarmi CloudWatch .

Argomenti

- Creazione di un AWS AppConfig ambiente (console)
- Creazione di un AWS AppConfig ambiente (riga di comando)

Creazione di un AWS AppConfig ambiente (console)

Utilizzare la procedura seguente per creare un AWS AppConfig ambiente utilizzando la AWS Systems Manager console.

Per creare un ambiente

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nella scheda Applicazioni, scegli il nome di un'applicazione per aprire la pagina dei dettagli.
3. Scegli la scheda Ambienti, quindi scegli Crea ambiente.
4. In Name (Nome), inserire un nome per l'ambiente.
5. In Description (Descrizione), immettere le informazioni sull'ambiente.
6. (Facoltativo) Nella sezione Monitor, scegli il campo del ruolo IAM, quindi scegli un ruolo IAM con il permesso di ripristinare una configurazione se viene attivato un allarme.
7. Nell'elenco degli CloudWatch allarmi, scegli uno o più allarmi da monitorare. AWS AppConfig ripristina l'implementazione della configurazione se uno di questi allarmi entra in uno stato di allarme.
8. (Facoltativo) Nella sezione Associa estensioni, scegli un'estensione dall'elenco. Per ulteriori informazioni, consulta Informazioni sulle AWS AppConfig estensioni.
9. (Facoltativo) Nella sezione Tag, inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
10. Seleziona Create environment (Crea ambiente).

AWS AppConfig crea l'ambiente e quindi visualizza la pagina dei dettagli dell'ambiente. Passa a Creazione di un profilo di configurazione in AWS AppConfig.

Creazione di un AWS AppConfig ambiente (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Tools for PowerShell creare un AWS AppConfig ambiente.

Creare un ambiente passo dopo passo

1. Apri il AWS CLI.
2. Eseguite il comando seguente per creare un ambiente.

   Linux

   ```
   aws appconfig create-environment \
     --application-id The_application_ID \
     --name A_name_for_the_environment \
     --description A_description_of_the_environment \
     --monitors
    "AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
    role_for_AWS AppConfig_to_monitor_AlarmArn" \
     --tags User_defined_key_value_pair_metadata_of_the_environment
   ```

   Windows

   ```
   aws appconfig create-environment ^
     --application-id The_application_ID ^
     --name A_name_for_the_environment ^
     --description A_description_of_the_environment ^
     --monitors
    "AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
    role_for_AWS AppConfig_to_monitor_AlarmArn" ^
     --tags User_defined_key_value_pair_metadata_of_the_environment
   ```

   PowerShell

   ```
   New-APPCEnvironment `
     -Name Name_for_the_environment `
     -ApplicationId The_application_ID
     -Description Description_of_the_environment `
   ```

```
  -Monitors
 @{"AlarmArn=ARN_of_the_Amazon_CloudWatch_alarm,AlarmArnRole=ARN_of_the_IAM
 role_for_AWS AppConfig_to_monitor_AlarmArn"} `
  -Tag Hashtable_type_user_defined_key_value_pair_metadata_of_the_environment
```

Il sistema restituisce informazioni simili alle seguenti.

Linux

```
{
   "ApplicationId": "The application ID",
   "Id": "The_environment ID",
   "Name": "Name of the environment",
   "State": "The state of the environment",
   "Description": "Description of the environment",

   "Monitors": [
      {
         "AlarmArn": "ARN of the Amazon CloudWatch alarm",
         "AlarmRoleArn": "ARN of the IAM role for AppConfig to monitor AlarmArn"
      }
   ]
}
```

Windows

```
{
   "ApplicationId": "The application ID",
   "Id": "The environment ID",
   "Name": "Name of the environment",
   "State": "The state of the environment"
   "Description": "Description of the environment",

   "Monitors": [
      {
         "AlarmArn": "ARN of the Amazon CloudWatch alarm",
         "AlarmRoleArn": "ARN of the IAM role for AppConfig to monitor AlarmArn"
      }
   ]
}
```

PowerShell

```
ApplicationId     : The application ID
ContentLength     : Runtime of the command
Description       : Description of the environment
HttpStatusCode    : HTTP Status of the runtime
Id                : The environment ID
Monitors          : {ARN of the Amazon CloudWatch alarm, ARN of the IAM role for
 AppConfig to monitor AlarmArn}
Name              : Name of the environment
Response Metadata : Runtime Metadata
State             : State of the environment
```

Passa a Creazione di un profilo di configurazione in AWS AppConfig.

Creazione di un profilo di configurazione in AWS AppConfig

Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di configurazione. AWS AppConfig supporta due tipi di profili di configurazione: flag di funzionalità e configurazioni a forma libera. Una configurazione con flag di funzionalità archivia i dati nell'archivio di configurazione AWS AppConfig ospitato e l'URI è semplice. `hosted` Una configurazione in formato libero può archiviare i dati nell'archivio di configurazione AWS AppConfig ospitato, in varie funzionalità di Systems Manager o in qualsiasi AWS servizio che si integra con. AWS AppConfig Per ulteriori informazioni, consulta Creazione di un profilo di configurazione in formato libero in AWS AppConfig.

Un profilo di configurazione può anche includere validatori opzionali per garantire che i dati di configurazione siano corretti dal punto di vista sintattico e semantico. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione si interrompe prima di apportare modifiche ai target di configurazione.

> **Note**
>
> Se possibile, consigliamo di ospitare i dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato in quanto offre la maggior parte delle funzionalità e dei miglioramenti.

Argomenti

Informazioni sui validatori

Quando crei un profilo di configurazione, hai la possibilità di specificare fino a due validatori. Un validatore assicura che i dati di configurazione siano sintatticamente e semanticamente corretti. Se intendete utilizzare un validatore, dovete crearlo prima di creare il profilo di configurazione. AWS AppConfig supporta i seguenti tipi di validatori:

- AWS Lambda funzioni: supportate per i flag delle funzionalità e le configurazioni in formato libero.
- Schema JSON: supportato per configurazioni in formato libero. (convalida AWS AppConfig automaticamente i flag delle funzionalità rispetto a uno schema JSON.)

Argomenti

AWS Lambda validatori di funzioni

I validatori delle funzioni Lambda devono essere configurati con il seguente schema di eventi. AWS AppConfig utilizza questo schema per richiamare la funzione Lambda. Il contenuto è una stringa con codifica base64 e l'uri è una stringa.

```
{
    "applicationId": "The application ID of the configuration profile being
 validated",
    "configurationProfileId": "The ID of the configuration profile being validated",
    "configurationVersion": "The version of the configuration profile being validated",
    "content": "Base64EncodedByteString",
    "uri": "The configuration uri"
}
```

AWS AppConfig verifica che l'intestazione `X-Amz-Function-Error` Lambda sia impostata nella risposta. Lambda imposta questa intestazione se la funzione genera un'eccezione. Per ulteriori informazioni in merito*`X-Amz-Function-Error`*, consulta la sezione Gestione degli errori e tentativi automatici AWS Lambda nella Guida per gli sviluppatori.AWS Lambda

Ecco un semplice esempio di codice di risposta Lambda per una convalida corretta.

```
import json

def handler(event, context):
     #Add your validation logic here
     print("We passed!")
```

Ecco un semplice esempio di codice di risposta Lambda per una convalida non riuscita.

```
def handler(event, context):
     #Add your validation logic here
     raise Exception("Failure!")
```

Ecco un altro esempio che convalida solo se il parametro di configurazione è un numero primo.

```
function isPrime(value) {
    if (value < 2) {
        return false;
    }

    for (i = 2; i < value; i++) {
        if (value % i === 0) {
            return false;
        }
    }

    return true;
}

exports.handler = async function(event, context) {
    console.log('EVENT: ' + JSON.stringify(event, null, 2));
    const input = parseInt(Buffer.from(event.content, 'base64').toString('ascii'));
    const prime = isPrime(input);
    console.log('RESULT: ' + input + (prime ? ' is' : ' is not') + ' prime');
    if (!prime) {
        throw input + "is not prime";
```

```
    }
}
```

AWS AppConfig chiama la tua Lambda di convalida quando richiama `StartDeployment` le operazioni `ValidateConfigurationActivity` e API. Devi fornire `appconfig.amazonaws.com` le autorizzazioni per richiamare la tua Lambda. Per ulteriori informazioni, consulta Concessione dell'accesso alle funzioni ai servizi. AWS AWS AppConfig limita il tempo di esecuzione Lambda di convalida a 15 secondi, inclusa la latenza di avvio.

validatori dello schema JSON

Se crei una configurazione in un documento SSM, devi specificare o creare uno schema JSON per la configurazione. Uno schema JSON definisce le proprietà consentite per ogni impostazione di configurazione dell'applicazione. Lo schema JSON funziona come un set di regole per garantire che le impostazioni di configurazione nuove o aggiornate siano conformi alle best practice richieste dall'applicazione. Ecco un esempio.

```
    {
      "$schema": "http://json-schema.org/draft-04/schema#",
      "title": "$id$",
      "description": "BasicFeatureToggle-1",
      "type": "object",
      "additionalProperties": false,
      "patternProperties": {
          "[^\\s]+$": {
              "type": "boolean"
          }
      },
      "minProperties": 1
    }
```

Quando si crea una configurazione da un documento SSM, il sistema verifica automaticamente che la configurazione sia conforme ai requisiti dello schema. In caso contrario, AWS AppConfig restituisce un errore di convalida.

> **Important**
>
> Nota le seguenti informazioni importanti sui validatori dello schema JSON:
>
> - I dati di configurazione archiviati nei documenti SSM devono essere convalidati rispetto a uno schema JSON associato prima di poter aggiungere la configurazione al sistema.

I parametri SSM non richiedono un metodo di convalida, ma si consiglia di creare un controllo di convalida per le configurazioni dei parametri SSM nuove o aggiornate utilizzando. AWS Lambda

- Una configurazione in un documento SSM utilizza il tipo di documento. `ApplicationConfiguration` Lo schema JSON corrispondente utilizza il tipo di `ApplicationConfigurationSchema` documento.
- AWS AppConfig supporta lo schema JSON versione 4.X per lo schema in linea. Se la configurazione dell'applicazione richiede una versione diversa di JSON Schema, devi creare un validatore Lambda.

Creazione di un profilo di configurazione del feature flag in AWS AppConfig

È possibile utilizzare i flag di funzionalità per abilitare o disabilitare le funzionalità all'interno delle applicazioni o per configurare caratteristiche diverse delle funzionalità dell'applicazione utilizzando gli attributi dei flag. AWS AppConfig archivia le configurazioni dei flag di funzionalità nell'archivio di configurazione AWS AppConfig ospitato in un formato di feature flag che contiene dati e metadati sui flag e sugli attributi dei flag. Per ulteriori informazioni sull'archivio di configurazione AWS AppConfig ospitato, consulta la sezione. Informazioni sull'archivio di configurazione AWS AppConfig ospitato

Argomenti

- Creazione di un feature flag e di un profilo di configurazione di feature flag (console)
- Creazione di un feature flag e di un profilo di configurazione del feature flag (riga di comando)
- Digita il riferimento per AWS.AppConfig.FeatureFlags

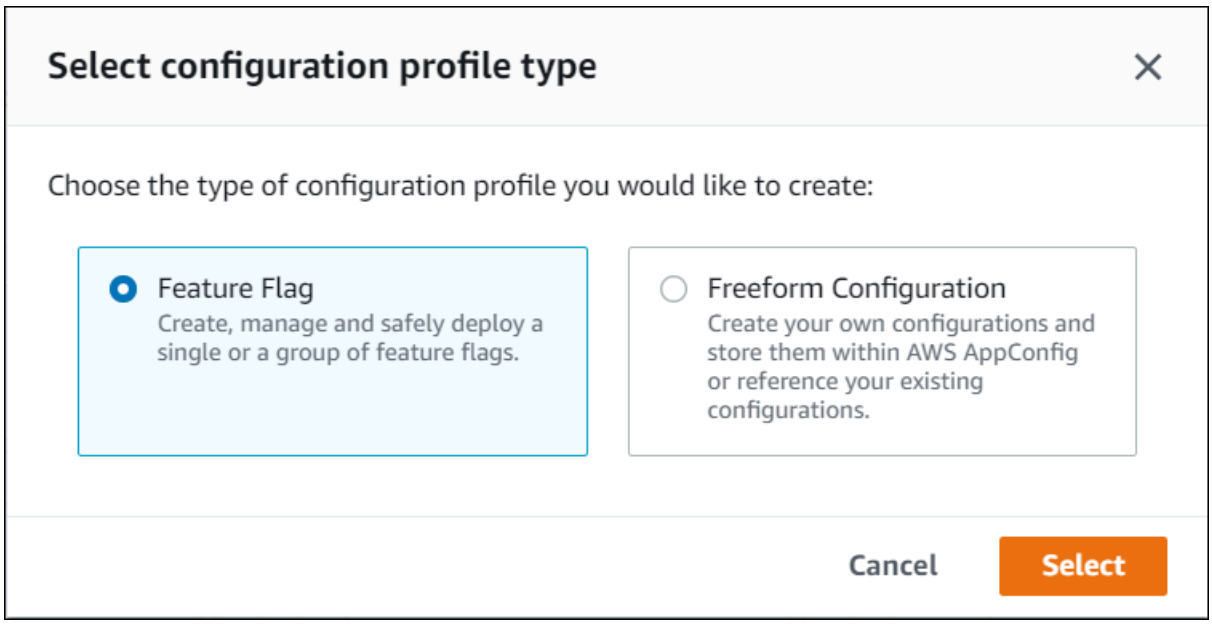
Creazione di un feature flag e di un profilo di configurazione di feature flag (console)

Utilizzare la procedura seguente per creare un profilo di configurazione dei AWS AppConfig feature flag e una configurazione dei feature flag utilizzando la AWS AppConfig console.

Per creare un profilo di configurazione

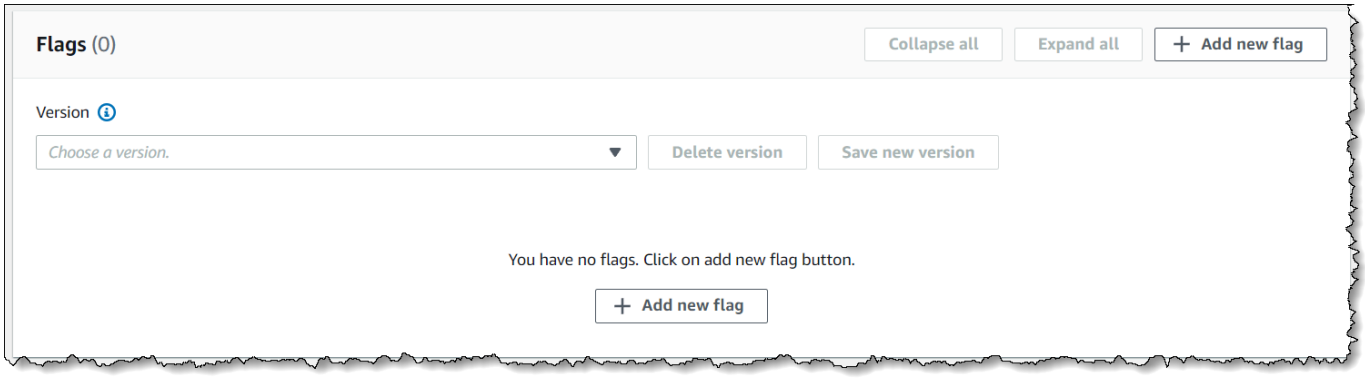
1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nella scheda Applicazioni, scegli l'applicazione che hai creato in Crea una AWS AppConfig configurazione, quindi scegli la scheda Profili di configurazione e bandiere di funzionalità.

3. Scegli Crea.
4. Scegli Feature flag.

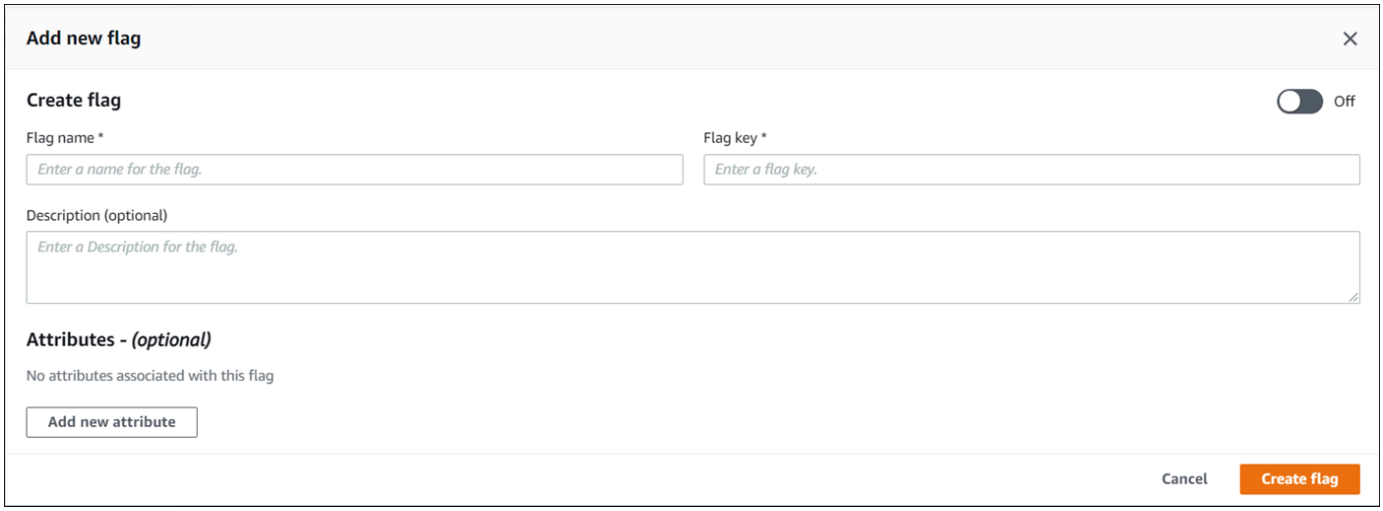


Per creare un flag di funzionalità

1. Nella configurazione che hai creato, scegli Aggiungi nuovo flag.



2. Fornisci un nome e una descrizione (opzionale) per la bandiera. Il tasto Flag si compila automaticamente sostituendo gli spazi con caratteri di sottolineatura nel nome fornito. Puoi modificare la chiave flag se desideri un valore o un formato diverso. Dopo aver creato il flag, è possibile modificare il nome del flag, ma non la chiave del flag.



3. Specificate se il flag della funzione è abilitato o disabilitato utilizzando il pulsante di attivazione/disattivazione.
4. (Facoltativo) Aggiungete attributi e vincoli di attributo al flag di funzionalità. Gli attributi consentono di fornire valori aggiuntivi all'interno del flag. Facoltativamente, è possibile convalidare i valori degli attributi in base a vincoli specifici. I vincoli assicurano che eventuali valori imprevisti non vengano distribuiti nell'applicazione.

   AWS AppConfig feature flags supporta i seguenti tipi di attributi e i vincoli corrispondenti.

Type	Vincolo	Descrizione
Stringa	Espressione regolare	Modello Regex per la stringa
	Enum	Elenco di valori accettabili per la stringa
Numero	Minimo	Valore numerico minimo per l'attributo
	Massimo	Valore numerico massimo per l'attributo
Booleano	Nessuno	Nessuno
Matrice di stringhe	Espressione regolare	Modello Regex per gli elementi dell'array

Type	Vincolo	Descrizione
	Enum	Elenco di valori accettabili per gli elementi dell'array
Matrice numerica	Minimo	Valore numerico minimo per gli elementi dell'array
	Massimo	Valore numerico massimo per gli elementi dell'array

> **Note**
>
> Osservare le seguenti informazioni.
>
> - Per i nomi degli attributi, la parola «abilitato» è riservata. Non è possibile creare un attributo di feature flag chiamato «enabled». Non ci sono altre parole riservate.
> - Gli attributi di un flag di funzionalità sono inclusi nella `GetLatestConfiguration` risposta solo se tale flag è abilitato.
> - Seleziona Valore richiesto per specificare se il valore di un attributo è obbligatorio.

5. Scegli Salva nuova versione.

Passa a Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig.

Creazione di un feature flag e di un profilo di configurazione del feature flag (riga di comando)

La procedura seguente descrive come utilizzare AWS Command Line Interface (su Linux o Windows) o Tools for Windows per PowerShell creare un profilo di configurazione dei AWS AppConfig feature flag. Se preferisci, puoi utilizzarlo AWS CloudShell per eseguire i comandi elencati di seguito. Per ulteriori informazioni, consulta Che cos'è AWS CloudShell? nella Guida per l'utente di AWS CloudShell .

Per creare una feature flag, la configurazione viene effettuata passo dopo passo

1. Apri il. AWS CLI

2. Crea un profilo di configurazione del feature flag specificandone il tipo come`AWS.AppConfig.FeatureFlags`. Il profilo di configurazione deve utilizzare `hosted` l'URI della posizione.

   Linux

   ```
   aws appconfig create-configuration-profile \
     --application-id The_application_ID \
     --name A_name_for_the_configuration_profile \
     --location-uri hosted \
     --type AWS.AppConfig.FeatureFlags
   ```

   Windows

   ```
   aws appconfig create-configuration-profile ^
     --application-id The_application_ID ^
     --name A_name_for_the_configuration_profile ^
     --location-uri hosted ^
     --type AWS.AppConfig.FeatureFlags
   ```

   PowerShell

   ```
   New-APPCConfigurationProfile `
     -Name A_name_for_the_configuration_profile `
     -ApplicationId The_application_ID `
     -LocationUri hosted `
     -Type AWS.AppConfig.FeatureFlags
   ```

3. Crea i dati di configurazione del tuo feature flag. I tuoi dati devono essere in formato JSON e conformi allo schema `AWS.AppConfig.FeatureFlags` JSON. Per ulteriori informazioni sullo schema, vedere. Digita il riferimento per AWS.AppConfig.FeatureFlags
4. Utilizza l'`CreateHostedConfigurationVersion`API per salvare i dati di configurazione del feature flag in AWS AppConfig.

   Linux

   ```
   aws appconfig create-hosted-configuration-version \
     --application-id The_application_ID \
     --configuration-profile-id The_configuration_profile_id \
   ```

```
  --content-type "application/json" \
  --content file://path/to/feature_flag_configuration_data \
  file_name_for_system_to_store_configuration_data
```

Windows

```
aws appconfig create-hosted-configuration-version ^
  --application-id The_application_ID ^
  --configuration-profile-id The_configuration_profile_id ^
  --content-type "application/json" ^
  --content file://path/to/feature_flag_configuration_data ^
  file_name_for_system_to_store_configuration_data
```

PowerShell

```
New-APPCHostedConfigurationVersion `
  -ApplicationId The_application_ID `
  -ConfigurationProfileId The_configuration_profile_id `
  -ContentType "application/json" `
  -Content file://path/to/feature_flag_configuration_data `
  file_name_for_system_to_store_configuration_data
```

Ecco un esempio di comando Linux.

```
aws appconfig create-hosted-configuration-version \
  --application-id 1a2b3cTestApp \
  --configuration-profile-id 4d5e6fTestConfigProfile \
  --content-type "application/json" \
  --content Base64Content
```

Il `content` parametro utilizza i seguenti dati `base64` codificati.

```
{
  "flags": {
    "flagkey": {
      "name": "WinterSpecialBanner"
    }
  },
  "values": {
    "flagkey": {
```

```
      "enabled": true
    }
  },
  "version": "1"
}
```

Il sistema restituisce informazioni simili alle seguenti.

Linux

```
{
   "ApplicationId"          : "1a2b3cTestApp",
   "ConfigurationProfileId" : "4d5e6fTestConfigProfile",
   "VersionNumber"          : "1",
   "ContentType"            : "application/json"
}
```

Windows

```
{
   "ApplicationId"          : "1a2b3cTestApp",
   "ConfigurationProfileId" : "4d5e6fTestConfigProfile",
   "VersionNumber"          : "1",
   "ContentType"            : "application/json"
}
```

PowerShell

```
ApplicationId          : 1a2b3cTestApp
ConfigurationProfileId : 4d5e6fTestConfigProfile
VersionNumber          : 1
ContentType            : application/json
```

`service_returned_content_file`Contiene i dati di configurazione che includono alcuni metadati AWS AppConfig generati.

> **Note**
>
> Quando crei la versione di configurazione ospitata, AWS AppConfig verifica che i dati siano conformi allo `AWS.AppConfig.FeatureFlags` schema JSON. AWS AppConfig

verifica inoltre che ogni attributo feature flag presente nei dati soddisfi i vincoli definiti per tali attributi.

Digita il riferimento per AWS.AppConfig.FeatureFlags

Utilizzate lo schema `AWS.AppConfig.FeatureFlags` JSON come riferimento per creare i dati di configurazione dei feature flag.

```
{
      "$schema": "http://json-schema.org/draft-07/schema#",
      "definitions": {
        "flagSetDefinition": {
          "type": "object",
          "properties": {
            "version": {
              "$ref": "#/definitions/flagSchemaVersions"
            },
            "flags": {
              "$ref": "#/definitions/flagDefinitions"
            },
            "values": {
              "$ref": "#/definitions/flagValues"
            }
          },
          "required": ["version", "flags"],
          "additionalProperties": false
        },
        "flagDefinitions": {
          "type": "object",
          "patternProperties": {
            "^[a-z][a-zA-Z\\d-]{0,63}$": {
              "$ref": "#/definitions/flagDefinition"
            }
          },
          "maxProperties": 100,
          "additionalProperties": false
        },
        "flagDefinition": {
          "type": "object",
          "properties": {
            "name": {
```

```
            "$ref": "#/definitions/customerDefinedName"
          },
          "description": {
            "$ref": "#/definitions/customerDefinedDescription"
          },
          "_createdAt": {
            "type": "string"
          },
          "_updatedAt": {
            "type": "string"
          },
          "_deprecation": {
            "type": "object",
            "properties": {
              "status": {
                "type": "string",
                "enum": ["planned"]
              }
            },
           "additionalProperties": false
          },
          "attributes": {
            "$ref": "#/definitions/attributeDefinitions"
          }
        },
        "additionalProperties": false
      },
      "attributeDefinitions": {
        "type": "object",
        "patternProperties": {
          "^[a-z][a-zA-Z\\d-_]{0,63}$": {
            "$ref": "#/definitions/attributeDefinition"
          }
        },
        "maxProperties": 25,
        "additionalProperties": false
      },
      "attributeDefinition": {
        "type": "object",
        "properties": {
          "description": {
            "$ref": "#/definitions/customerDefinedDescription"
          },
          "constraints": {
```

```
            "oneOf": [
              { "$ref": "#/definitions/numberConstraints" },
              { "$ref": "#/definitions/stringConstraints" },
              { "$ref": "#/definitions/arrayConstraints" },
              { "$ref": "#/definitions/boolConstraints" }
            ]
          }
        },
        "additionalProperties": false
      },
      "flagValues": {
        "type": "object",
        "patternProperties": {
          "^[a-z][a-zA-Z\\d-_]{0,63}$": {
            "$ref": "#/definitions/flagValue"
          }
        },
        "maxProperties": 100,
        "additionalProperties": false
      },
      "flagValue": {
        "type": "object",
        "properties": {
          "enabled": {
            "type": "boolean"
          },
          "_createdAt": {
            "type": "string"
          },
          "_updatedAt": {
            "type": "string"
          }
        },
        "patternProperties": {
          "^[a-z][a-zA-Z\\d-_]{0,63}$": {
            "$ref": "#/definitions/attributeValue",
            "maxProperties": 25
          }
        },
        "required": ["enabled"],
        "additionalProperties": false
      },
      "attributeValue": {
        "oneOf": [
```

```
            { "type": "string", "maxLength": 1024 },
            { "type": "number" },
            { "type": "boolean" },
            {
              "type": "array",
              "oneOf": [
                {
                  "items": {
                    "type": "string",
                    "maxLength": 1024
                  }
                },
                {
                  "items": {
                    "type": "number"
                  }
                }
              ]
            }
          ],
          "additionalProperties": false
        },
        "stringConstraints": {
          "type": "object",
          "properties": {
            "type": {
              "type": "string",
              "enum": ["string"]
      },
            "required": {
              "type": "boolean"
            },
            "pattern": {
              "type": "string",
              "maxLength": 1024
            },
            "enum": {
              "type": "array",
              "maxLength": 100,
              "items": {
                "oneOf": [
                  {
                    "type": "string",
                    "maxLength": 1024
```

```
                },
                {
                  "type": "integer"
                }
              ]
            }
          }
        },
        "required": ["type"],
        "not": {
          "required": ["pattern", "enum"]
        },
        "additionalProperties": false
      },
      "numberConstraints": {
        "type": "object",
        "properties": {
          "type": {
            "type": "string",
            "enum": ["number"]
    },
          "required": {
            "type": "boolean"
          },
          "minimum": {
            "type": "integer"
          },
          "maximum": {
            "type": "integer"
          }
        },
        "required": ["type"],
        "additionalProperties": false
      },
      "arrayConstraints": {
        "type": "object",
        "properties": {
          "type": {
            "type": "string",
            "enum": ["array"]
    },
          "required": {
            "type": "boolean"
          },
```

```
            "elements": {
              "$ref": "#/definitions/elementConstraints"
      }
          },
          "required": ["type"],
          "additionalProperties": false
        },
        "boolConstraints": {
          "type": "object",
          "properties": {
            "type": {
              "type": "string",
              "enum": ["boolean"]
      },
            "required": {
              "type": "boolean"
            }
          },
          "required": ["type"],
          "additionalProperties": false
        },
        "elementConstraints": {
          "oneOf": [
            { "$ref": "#/definitions/numberConstraints" },
            { "$ref": "#/definitions/stringConstraints" }
          ]
        },
        "customerDefinedName": {
          "type": "string",
          "pattern": "^[^\\n]{1,64}$"
        },
        "customerDefinedDescription": {
          "type": "string",
          "maxLength": 1024
        },
        "flagSchemaVersions": {
          "type": "string",
          "enum": ["1"]
        }
      },
      "type": "object",
      "$ref": "#/definitions/flagSetDefinition",
      "additionalProperties": false
```

```
    }
```

> **Important**
>
> Per recuperare i dati di configurazione dei feature flag, l'applicazione deve chiamare l'`GetLatestConfiguration`API. Non è possibile recuperare i dati di configurazione dei feature flag chiamando`GetConfiguration`, il che è obsoleto. Per ulteriori informazioni, consulta l'API Reference GetLatestConfiguration.AWS AppConfig

Quando l'applicazione chiama GetLatestConfiguratione riceve una configurazione appena distribuita, le informazioni che definiscono i flag e gli attributi delle funzionalità vengono rimosse. Il JSON semplificato contiene una mappa di chiavi che corrispondono a ciascuna delle chiavi flag specificate. Il JSON semplificato contiene anche i valori mappati di `true` o per l'attributo. `false` `enabled` Se un flag è `enabled` impostato su`true`, saranno presenti anche tutti gli attributi del flag. Lo schema JSON seguente descrive il formato dell'output JSON.

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "type": "object",
  "patternProperties": {
    "^[a-z][a-zA-Z\\d-_]{0,63}$": {
      "$ref": "#/definitions/attributeValuesMap"
    }
  },
  "maxProperties": 100,
  "additionalProperties": false,
  "definitions": {
    "attributeValuesMap": {
      "type": "object",
      "properties": {
        "enabled": {
          "type": "boolean"
        }
      },
      "required": ["enabled"],
      "patternProperties": {
        "^[a-z][a-zA-Z\\d-_]{0,63}$": {
          "$ref": "#/definitions/attributeValue"
        }
      },
```

```
      "maxProperties": 25,
      "additionalProperties": false
    },
    "attributeValue": {
      "oneOf": [
        { "type": "string","maxLength": 1024 },
        { "type": "number" },
        { "type": "boolean" },
        {
          "type": "array",
          "oneOf": [
            {
              "items": {
                "oneOf": [
                  {
                    "type": "string",
                    "maxLength": 1024
                  }
                ]
              }
            },
            {
              "items": {
                "oneOf": [
                  {
                    "type": "number"
                  }
                ]
              }
            }
          ]
        }
      ],
      "additionalProperties": false
    }
  }
}
```

Creazione di un profilo di configurazione in formato libero in AWS AppConfig

Un profilo di configurazione include, tra le altre cose, un URI che consente di AWS AppConfig localizzare i dati di configurazione nella posizione archiviata e un tipo di profilo. AWS AppConfig supporta due tipi di profili di configurazione: flag di funzionalità e configurazioni a forma libera. I profili di configurazione Feature Flag archiviano i propri dati nell'archivio di configurazione AWS AppConfig ospitato e l'URI è semplice. `hosted` Per i profili di configurazione in formato libero, è possibile archiviare i dati nell'archivio di configurazione AWS AppConfig ospitato o in uno dei seguenti AWS servizi e funzionalità di Systems Manager:

Ubicazione	Tipi di file supportati
AWS AppConfig archivio di configurazione ospitato	YAML, JSON e testo se aggiunti utilizzando. AWS Management Console Qualsiasi tipo di file se aggiunto utilizzando l' AWS AppConfig CreateHostedConfigurationVersionazione API.
Amazon Simple Storage Service (Amazon S3)	Qualsiasi
AWS CodePipeline	Pipeline (come definita dal servizio)
AWS Secrets Manager	Segreto (come definito dal servizio)
AWS Systems Manager Archivio parametri	Parametri di stringa standard e sicuri (come definiti da Parameter Store)
AWS Systems Manager archivio documenti (documenti SSM)	YAML, JSON, testo

Un profilo di configurazione può anche includere validatori opzionali per garantire che i dati di configurazione siano corretti dal punto di vista sintattico e semantico. AWS AppConfig esegue un controllo utilizzando i validatori quando si avvia una distribuzione. Se vengono rilevati errori, la distribuzione si interrompe prima di apportare modifiche ai target di configurazione.

> **Note**
>
> Se possibile, consigliamo di ospitare i dati di configurazione nell'archivio di configurazione AWS AppConfig ospitato in quanto offre la maggior parte delle funzionalità e dei miglioramenti.

Per le configurazioni in formato libero archiviate nell'archivio di configurazione AWS AppConfig ospitato o nei documenti SSM, è possibile creare la configurazione in formato libero utilizzando la console Systems Manager al momento della creazione di un profilo di configurazione. Il processo è descritto più avanti in questo argomento.

Per le configurazioni in formato libero archiviate in Parameter Store, Secrets Manager o Amazon S3, devi prima creare il parametro, il segreto o l'oggetto e archiviarlo nell'archivio di configurazione pertinente. Dopo aver archiviato i dati di configurazione, utilizza la procedura descritta in questo argomento per creare il profilo di configurazione.

Argomenti

Informazioni sulle quote e limitazioni dell'archivio di configurazione

Gli archivi di configurazione supportati da AWS AppConfig hanno le quote e le limitazioni seguenti.

	AWS AppConfig archivio di configurazione ospitato	Amazon S3	Systems Manager Parameter Store	AWS Secrets Manager	Systems Manager Archivio documenti	AWS CodePipeline
Limite di dimension	2 MB per impostazi	2 MB	4 KB (piano	64 KB	64 KB	2 MB

	AWS AppConfig archivio di configurazione ospitato	Amazon S3	Systems Manager Parameter Store	AWS Secrets Manager	Systems Manager Archivio documenti	AWS CodePipeline
e della configurazione	one predefinita, 4 MB massimi	Applicato da AWS AppConfig, non da S3	gratuito) /8 KB (parametri avanzati)			Applicato da, non AWS AppConfig CodePipeline
Limite di storage delle risorse	1 GB	Illimitato	10.000 parametri (piano gratuito) /100.000 parametri (parametri avanzati)	500.000	500 documenti	Limitato dal numero di profili di configurazione per applicazione (100 profili per applicazione)
Crittografia lato server	Sì	SSE-S3, SSE-KMS	Sì	Sì	No	Sì
AWS CloudFormation supporto	Sì	Non per la creazione o l'aggiornamento dei dati	Sì	Sì	No	Sì

	AWS AppConfig archivio di configurazione ospitato	Amazon S3	Systems Manager Parameter Store	AWS Secrets Manager	Systems Manager Archivio documenti	AWS CodePipeline
Prezzi	Gratuito	Vedi i prezzi di Amazon S3	Vedi i prezzi AWS Systems Manager	Vedi AWS Secrets Manager i prezzi	Gratuito	Vedi AWS CodePipeline i prezzi

Informazioni sull'archivio di configurazione AWS AppConfig ospitato

AWS AppConfig include un archivio di configurazione interno o ospitato. Le configurazioni devono pesare almeno 2 MB. L'archivio di configurazione AWS AppConfig ospitato offre i seguenti vantaggi rispetto ad altre opzioni dell'archivio di configurazione.

- Non è necessario impostare e configurare altri servizi come Amazon Simple Storage Service (Amazon S3) o Archivio parametri.
- Non è necessario configurare le autorizzazioni AWS Identity and Access Management (IAM) per utilizzare l'archivio di configurazione.
- Puoi memorizzare le configurazioni in YAML, JSON o come documenti di testo.
- L'utilizzo dell'archivio è gratuito.
- Puoi creare una configurazione e aggiungerla all'archivio quando crei un profilo di configurazione.

Informazioni sulle configurazioni archiviate in Amazon S3

Puoi archiviare le configurazioni in un bucket Amazon Simple Storage Service (Amazon S3). Quando si crea il profilo di configurazione, si specifica l'URI di un singolo oggetto S3 in un bucket. È inoltre necessario specificare l'Amazon Resource Name (ARN) di un ruolo AWS Identity and Access Management (IAM) che AWS AppConfig autorizza a ottenere l'oggetto. Prima di creare un profilo di configurazione per un oggetto Amazon S3, tieni presente le seguenti restrizioni.

Limitazione	Informazioni
Size	Le configurazioni archiviate come oggetti S3 possono avere una dimensione massima di 1 MB.
Crittografia degli oggetti	Un profilo di configurazione può avere come target oggetti crittografati SSE-S3 e SSE-KMS.
Classi di archiviazione	AWS AppConfig supporta le seguenti classi di storage S3:,,, e. `STANDARD` `INTELLIGENT_TIERING` `REDUCED_REDUNDANCY` `STANDARD_IA` `ONEZONE_IA` Le seguenti classi non sono supportate: tutte le classi S3 Glacier (`GLACIER` e `DEEP_ARCHIVE`).
Controllo delle versioni	AWS AppConfig richiede che l'oggetto S3 utilizzi il controllo delle versioni.

Configurazione delle autorizzazioni per una configurazione archiviata come oggetto Amazon S3

Quando crei un profilo di configurazione per una configurazione archiviata come oggetto S3, devi specificare un ARN per un ruolo IAM che AWS AppConfig dia il permesso di ottenere l'oggetto. Il ruolo deve includere le seguenti autorizzazioni:

Autorizzazioni per accedere all'oggetto S3

- s3: GetObject
- s3: GetObjectVersion

Autorizzazioni per elencare i bucket S3

s3: ListAllMyBuckets

Autorizzazioni per accedere al bucket S3 in cui è archiviato l'oggetto

- s3: GetBucketLocation
- s3: GetBucketVersioning

- s3: ListBucket
- s3: ListBucketVersions

Completa la seguente procedura per creare un ruolo che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

Creazione della policy IAM per l'accesso a un oggetto S3

Utilizza la seguente procedura per creare una policy IAM che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

Per creare una policy IAM per l'accesso a un oggetto S3

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
3. Nella pagina Crea policy, scegli la scheda JSON.
4. Aggiornare la policy di esempio riportata di seguito con informazioni sul bucket S3 e sull'oggetto di configurazione. Quindi incollare il criterio nel campo di testo nella scheda JSON .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion"
      ],
      "Resource": "arn:aws:s3:::my-bucket/my-configurations/my-configuration.json"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetBucketVersioning",
        "s3:ListBucketVersions",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::my-bucket"
      ]
```

```
    },
    {
      "Effect": "Allow",
      "Action": "s3:ListAllMyBuckets",
      "Resource": "*"
    }
  ]
}
```

5. Scegli Verifica policy.
6. Nella pagina Review (Revisione) immettere un nome nella casella Name (Nome), quindi digitare una descrizione.
7. Scegli Crea policy. Il sistema ti riporta alla pagina Ruoli.

Creazione del ruolo IAM per l'accesso a un oggetto S3

Utilizza la seguente procedura per creare un ruolo IAM che AWS AppConfig consenta di archiviare una configurazione in un oggetto S3.

Per creare un ruolo IAM per l'accesso a un oggetto Amazon S3

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli e quindi Crea ruolo.
3. Nella sezione Seleziona il tipo di entità affidabile, scegli il AWS servizio.
4. Nella sezione Scegli un caso d'uso in Casi d'uso comuni, scegliere EC2, e quindi scegliere Next: Permissions.
5. Nella pagina Allega criteri autorizzazioni, nella casella di ricerca, immettere il nome del criterio creato nella procedura precedente.
6. Scegliere questa policy, quindi selezionare Next: Tags.
7. Nella pagina Aggiungi tag (opzionale), inserisci una chiave e un valore opzionale, quindi scegli Avanti: revisione.
8. Nella pagina Review (Revisione) immettere un nome nella casella Role name (Nome ruolo), quindi digitare una descrizione.
9. Scegliere Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Roles (Ruoli).
10. Nella pagina Roles (Ruoli) scegliere il ruolo appena creato per aprire la pagina Summary (Riepilogo). Annotare i valori per Role Name (Nome ruolo) e Role ARN (ARN ruolo). Verrà specificato il ruolo ARN quando si crea il profilo di configurazione più avanti in questo argomento.

Creazione di una relazione di trust

Utilizza la procedura seguente per configurare il ruolo appena creato per considerare attendibile AWS AppConfig.

Per aggiungere una relazione di trust

1. Nella pagina Summary (Riepilogo) per il ruolo creato in precedenza, scegliere la scheda Trust Relationships (Relazioni di trust), quindi scegliere Edit Trust Relationship (Modifica relazione di trust).
2. Eliminare `"ec2.amazonaws.com"` e aggiungere `"appconfig.amazonaws.com"` come mostrato nell'esempio seguente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appconfig.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

3. Scegli Update Trust Policy (Aggiorna policy di trust).

Creazione di una configurazione e di un profilo di configurazione in formato libero

Questa sezione descrive come creare una configurazione e un profilo di configurazione in formato libero. Prima di iniziare, prendete nota delle seguenti informazioni.

- La procedura seguente richiede di specificare un ruolo del servizio IAM in modo che AWS AppConfig possa accedere ai dati di configurazione nell'archivio di configurazione scelto. Questo ruolo non è richiesto se si utilizza l'archivio di configurazione AWS AppConfig ospitato. Se scegli S3, Parameter Store o l'archivio documenti Systems Manager, devi scegliere un ruolo IAM esistente o scegliere l'opzione per fare in modo che il sistema crei automaticamente il ruolo per te. Per ulteriori informazioni su questo ruolo, consulta Informazioni sul profilo di configurazione (ruolo IAM).

- Per creare un profilo di configurazione per le configurazioni archiviate in S3, devi configurare le autorizzazioni. Per ulteriori informazioni sulle autorizzazioni e altri requisiti per l'utilizzo di S3 come archivio di configurazione, consulta Informazioni sulle configurazioni archiviate in Amazon S3.
- Se desideri utilizzare i validatori, esamina i dettagli e i requisiti per il loro utilizzo. Per ulteriori informazioni, consulta Informazioni sui validatori.

Argomenti

Creazione di un profilo di configurazione in AWS AppConfig formato libero (console)

Utilizzare la procedura seguente per creare un profilo di configurazione a AWS AppConfig forma libera e (facoltativamente) una configurazione a forma libera utilizzando la console. AWS Systems Manager

Per creare un profilo di configurazione

1. Aprire la console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/. AWS Systems Manager
2. Nella scheda Applicazioni, scegliete un'applicazione, quindi scegliete la scheda Profili di configurazione e bandiere di funzionalità.
3. Scegli Crea.
4. Scegliete Configurazione in formato libero, quindi selezionate Seleziona.
5. Per Name (Nome), immettere un nome per il profilo di configurazione.
6. Per Description (Descrizione), immettere informazioni sul profilo di configurazione.
7. Nella sezione Origine della configurazione, scegli un'opzione.
8.
   - Se hai selezionato la configurazione AWS AppConfig ospitata, scegli Text, JSON o YAML e inserisci la configurazione nel campo. Scegli Successivo e vai alla fase 10 di questa procedura.
   - Se hai selezionato un oggetto Amazon S3, inserisci l'URI dell'oggetto nel campo S3 object source, quindi scegli Avanti.
   - Se hai selezionato un AWS Systems Manager parametro, scegli il nome del parametro dall'elenco. Seleziona Successivo.

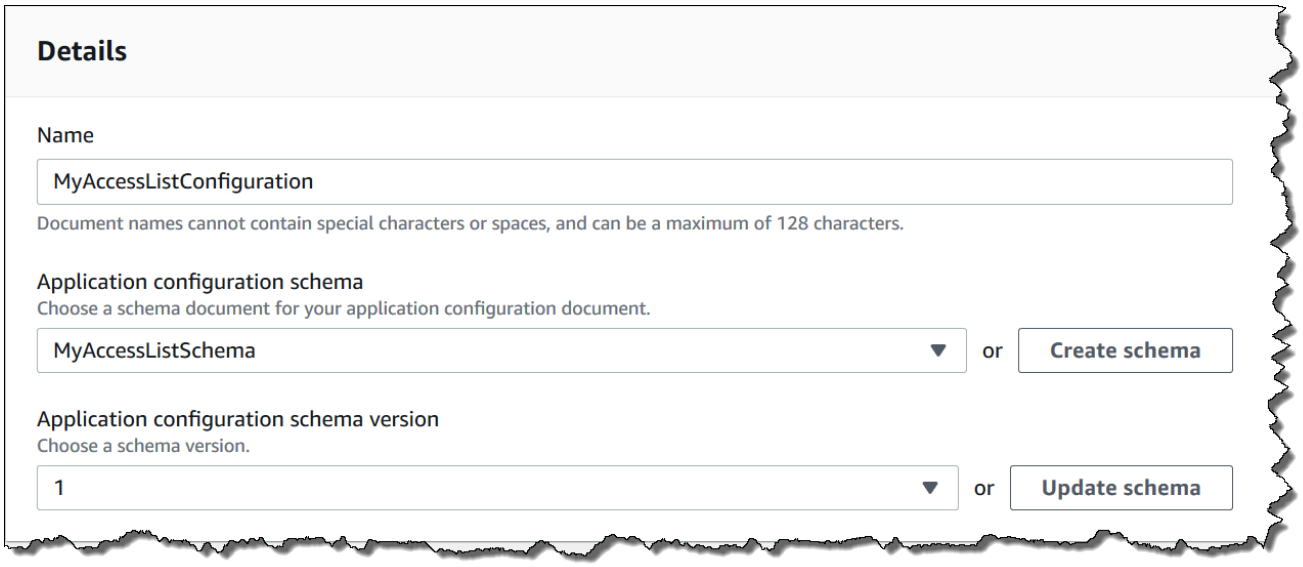
- Se hai selezionato Secrets Manager secret, inserisci il nome del segreto. Seleziona Successivo.
- Se hai selezionato AWS CodePipeline, scegli Avanti e vai al passaggio 10 di questa procedura.
- Se hai selezionato AWS Systems Manager un documento, completa i seguenti passaggi.

a. Nella sezione Origine documento scegliere Documento salvato o Nuovo documento.

b. Se scegli Documento salvato, scegli il documento SSM dall'elenco. Se si sceglie Nuovo documento, vengono visualizzate le sezioni Dettagli e Contenuto .

c. Nella sezione Dettagli immettere un nome per la nuova configurazione dell'applicazione.

d. Per la sezione Schema di configurazione dell'applicazione scegliere lo schema JSON utilizzando l'elenco o scegliere Crea schema. Se si sceglie Crea schema, Systems Manager apre la pagina Crea schema. Immettere i dettagli dello schema nella sezione Contenuto, quindi scegliere Crea schema.



e. Per la versione dello schema di configurazione dell'applicazione scegliere la versione dall'elenco o scegliere Aggiorna schema per modificare lo schema e creare una nuova versione.

f. Nella sezione Contenuto scegliere YAML o JSON e quindi immettere i dati di configurazione nel campo.

g. Seleziona Successivo.

9. Nella sezione Ruolo di servizio, scegli Nuovo ruolo di servizio per AWS AppConfig creare il ruolo IAM che fornisce l'accesso ai dati di configurazione. AWS AppConfig compila automaticamente

il campo Nome del ruolo in base al nome inserito in precedenza. Oppure, per scegliere un ruolo già esistente in IAM, scegli Ruolo di servizio esistente. Scegliere il ruolo utilizzando l'elenco Role ARN (Arn ruolo).

10. Nella pagina Add validators (Aggiungi validatori), scegliere Schema JSON o AWS Lambda. Se si sceglie JSON Scheme (Schema JSON), immettere lo schema JSON nel campo. Se scegli AWS Lambda, scegli la funzione Amazon Resource Name (ARN) e la versione dall'elenco.

    > **Important**
    >
    > I dati di configurazione archiviati nei documenti SSM devono essere convalidati rispetto a uno schema JSON associato prima di poter aggiungere la configurazione al sistema. I parametri SSM non richiedono un metodo di convalida, ma ti consigliamo di creare un controllo di convalida per configurazioni di parametri SSM nuove o aggiornate utilizzando. AWS Lambda

11. (Facoltativo) Nella sezione Tag, inserisci una chiave e un valore opzionale. È possibile specificare un massimo di 50 tag per una risorsa.
12. Scegli Crea set di configurazione.

> **Important**
>
> Se hai creato un profilo di configurazione per AWS CodePipeline, dopo aver creato una strategia di distribuzione, come descritto nella sezione successiva, devi creare una pipeline CodePipeline che indichi AWS AppConfig come provider di distribuzione. Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di distribuzione, vedi Tutorial: Create a pipeline that Uses AWS AppConfig as a Deployment Provider nella Guida per l'utente.AWS CodePipeline

Passa a Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig.

Creazione di un profilo di configurazione in AWS AppConfig formato libero (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Tools for PowerShell creare un profilo di configurazione in AWS AppConfig formato libero. Se preferisci, puoi usare AWS CloudShell per eseguire i comandi elencati di seguito. Per ulteriori informazioni, consulta Che cos'è AWS CloudShell? nella Guida per l'utente di AWS CloudShell .

> **Note**
>
> Per le configurazioni in formato libero ospitate nell'archivio di configurazione AWS AppConfig ospitato, è necessario specificare l'`hosted`URI di posizione.

Per creare un profilo di configurazione passo dopo passo

1. Apri il AWS CLI.
2. Esegui il comando seguente per creare un profilo di configurazione in formato libero.

   Linux

   ```
   aws appconfig create-configuration-profile \
     --application-id The_application_ID \
     --name A_name_for_the_configuration_profile \
     --description A_description_of_the_configuration_profile \
     --location-uri A_URI_to_locate_the_configuration or hosted \
     --retrieval-role-
   arn The_ARN_of_the_IAM_role_with_permission_to_access_the_configuration_at_the_specified
    \
     --tags User_defined_key_value_pair_metadata_of_the_configuration_profile \
     --validators "Content=JSON_Schema_content_or_the_ARN_of_an_AWS
    Lambda_function,Type=JSON_SCHEMA or LAMBDA"
   ```

   Windows

   ```
   aws appconfig create-configuration-profile ^
     --application-id The_application_ID ^
     --name A_name_for_the_configuration_profile ^
     --description A_description_of_the_configuration_profile ^
     --location-uri A_URI_to_locate_the_configuration or hosted  ^
     --retrieval-role-
   arn The_ARN_of_the_IAM_role_with_permission_to_access_the_configuration_at_the_specified
    ^
     --tags User_defined_key_value_pair_metadata_of_the_configuration_profile ^
     --validators "Content=JSON_Schema_content_or_the_ARN_of_an_AWS
    Lambda_function,Type=JSON_SCHEMA or LAMBDA"
   ```

PowerShell

```
New-APPCConfigurationProfile `
  -Name A_name_for_the_configuration_profile `
  -ApplicationId The_application_ID `
  -Description Description_of_the_configuration_profile `
  -LocationUri A_URI_to_locate_the_configuration or hosted `
  -
RetrievalRoleArn The_ARN_of_the_IAM_role_with_permission_to_access_the_configuration_at_
 `
  -
Tag Hashtable_type_user_defined_key_value_pair_metadata_of_the_configuration_profile
 `
  -Validators "Content=JSON_Schema_content_or_the_ARN_of_an_AWS
 Lambda_function,Type=JSON_SCHEMA or LAMBDA"
```

> **Note**
>
> Se hai creato una configurazione nell'archivio di configurazione AWS AppConfig ospitato, puoi creare nuove versioni della configurazione utilizzando le operazioni CreateHostedConfigurationVersionAPI. Per visualizzare AWS CLI i dettagli e i comandi di esempio per questa operazione API, consulta create-hosted-configuration-versionla sezione AWS CLI Command Reference.

Altre fonti di dati di configurazione

Questo argomento include informazioni su altri AWS servizi che si integrano con AWS AppConfig.

AWS AppConfig integrazione con AWS Secrets Manager

Secrets Manager consente di crittografare, archiviare e recuperare in modo sicuro le credenziali per i database e altri servizi. Invece di inserire le credenziali nelle app, puoi effettuare chiamate a Secrets Manager per recuperare le credenziali ogni volta che ne hai bisogno. Secrets Manager ti aiuta a proteggere l'accesso alle risorse e ai dati IT consentendoti di ruotare e gestire l'accesso ai tuoi segreti.

Quando si crea un profilo di configurazione in formato libero, è possibile scegliere Secrets Manager come origine dei dati di configurazione. È necessario effettuare l'onboarding con Secrets Manager e creare un segreto prima di creare il profilo di configurazione. Per ulteriori informazioni su Secrets Manager, vedi Cos'è AWS Secrets Manager? nella Guida AWS Secrets Manager per l'utente.
Per informazioni sulla creazione di un profilo di configurazione che utilizza Secrets Manager, vedereCreazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig.

Distribuzione di flag di funzionalità e dati di configurazione in AWS AppConfig

Dopo aver creato gli elementi necessari per l'utilizzo dei flag di funzionalità e dei dati di configurazione in formato libero, è possibile creare una nuova distribuzione. Quando si crea una nuova distribuzione, si specificano le seguenti informazioni:

- Un ID dell'applicazione
- Un ID del profilo di configurazione
- Una versione di configurazione
- Un ID di ambiente in cui si desidera distribuire i dati di configurazione
- Un ID della strategia di implementazione che definisce la velocità con cui desiderate che le modifiche abbiano effetto
- Un ID chiave AWS Key Management Service (AWS KMS) per crittografare i dati utilizzando una chiave gestita dal cliente.

Quando richiami l'azione StartDeploymentAPI, AWS AppConfig esegue le seguenti attività:

1. Recupera i dati di configurazione dal data store sottostante utilizzando l'URI di posizione nel profilo di configurazione.
2. Verifica che i dati di configurazione siano corretti dal punto di vista sintattico e semantico utilizzando i validatori specificati al momento della creazione del profilo di configurazione.
3. Memorizza nella cache una copia dei dati in modo che sia pronta per essere recuperata dall'applicazione. Questa copia memorizzata nella cache è denominata dati distribuiti.

AWS AppConfigsi integra con Amazon CloudWatch per monitorare le distribuzioni. Se una distribuzione attiva un allarme CloudWatch, ripristina AWS AppConfig automaticamente la distribuzione per ridurre al minimo l'impatto sugli utenti dell'applicazione.

Argomenti

- Utilizzo delle strategie di implementazione
- Distribuzione di una configurazione
- AWS AppConfigimplementazione, integrazione con CodePipeline

Utilizzo delle strategie di implementazione

Una strategia di implementazione consente di rilasciare lentamente le modifiche agli ambienti di produzione nell'arco di minuti o ore. Una strategia AWS AppConfig di implementazione definisce i seguenti aspetti importanti di una distribuzione di configurazione.

<table>
<tr><th>Impostazione</th><th>Descrizione</th></tr>
<tr><td>Il tipo di distribuzione</td><td>Il tipo di distribuzione definisce il modo in cui la configurazione viene distribuita. AWS AppConfig supporta i tipi di distribuzione lineare ed esponenziale .

• Lineare: per questo tipo, AWS AppConfig elabora l'implementazione mediante incrementi del fattore di crescita distribuiti uniformemente sull'implementazione. Ecco un esempio di cronologia per un'implem entazione di 10 ore che utilizza una crescita lineare del 20%:

<table>
<tr><th>Tempo trascorso</th><th>Avanzamento della distribuzione</th></tr>
<tr><td>0 ore</td><td>0%</td></tr>
<tr><td>2 ore</td><td>20%</td></tr>
<tr><td>4 ore</td><td>40%</td></tr>
<tr><td>6 ore</td><td>60%</td></tr>
<tr><td>8 ore</td><td>80%</td></tr>
<tr><td>10 ore</td><td>100%</td></tr>
</table>
• Esponenziale: per questo tipo, AWS AppConfig elabora la distribuzione in modo esponenziale utilizzando la seguente</td></tr>
</table>

Impostazione	Descrizione

	formula: `G*(2^N)`. In questa formula, `G` è la percentuale di fasi specificata dall'utente e `N` è il numero di fasi fino a quando la configurazione viene distribuita a tutte le destinazioni. Ad esempio, se si specifica un fattore di crescita di 2, il sistema esegue la configurazione come segue:
	`2*(2^0)`
	`2*(2^1)`
	`2*(2^2)`
	Espressa numericamente, la distribuzione viene eseguita come segue: 2% degli obiettivi, 4% degli obiettivi, 8% degli obiettivi e continua fino a quando la configurazione non è stata distribuita su tutti i target.

Percentuale di fasi (fattore di crescita)	Questa impostazione specifica la percentuale di chiamanti da destinare durante ogni fase della distribuzione **Note** Nell'SDK e nella Documentazione di riferimento delle API AWS AppConfig, `step percentage` viene chiamato `growth factor`.
Tempo di distribuzione	Questa impostazione specifica un periodo di tempo durante il quale AWS AppConfig viene distribuito agli host. Questo non è un valore di timeout. Si tratta di una finestra temporale durante la quale la distribuzione viene elaborata a intervalli.

Impostazione	Descrizione
Tempo	Questa impostazione specifica il periodo di tempo per il AWS AppConfig monitoraggio degli CloudWatch allarmi Amazon dopo che la configurazione è stata distribuita al 100% dei relativi obiettivi, prima di considerarla completa. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione. È necessario configurare le autorizzazioni per eseguire il rollback in base AWS AppConfig agli allarmi. CloudWatch Per ulteriori informazioni, consulta (Facoltativo) Configura le autorizzazioni per il rollback in base agli allarmi CloudWatch .

Puoi scegliere una strategia predefinita inclusa AWS AppConfig o crearne una personalizzata.

Argomenti

- Strategie di distribuzione predefinite
- Creazione di una strategia di distribuzione

Strategie di distribuzione predefinite

AWS AppConfig include le strategie di distribuzione predefinite che ti consentono di distribuire rapidamente una configurazione. Anziché creare strategie personalizzate, puoi scegliere una delle seguenti opzioni quando distribuisci una configurazione.

Strategia di distribuzione	Descrizione
AppConfigPercentEvery.Lineare 20 6 minuti	AWSconsigliato: Questa strategia implementa la configurazione sul 20% di tutti gli obiettivi ogni sei minuti per un'implementazione di 30 minuti. Il sistema monitora gli CloudWatch allarmi Amazon per

Strategia di distribuzione	Descrizione
	30 minuti. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione. Consigliamo di utilizzare questa strategia per le implementazioni di produzione perché è in linea con le AWS migliori pratiche e include un'enfasi aggiuntiva sulla sicurezza dell'implementazio ne grazie alla sua lunga durata e ai tempi di cottura.
AppConfig. 10 percento e 20 minuti	AWSconsigliato: questa strategia elabora la distribuzione in modo esponenziale utilizzando un fattore di crescita del 10% in 20 minuti. Il sistema monitora la presenza di CloudWatch allarmi per 10 minuti. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completat a. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione. Consigliamo di utilizzare questa strategia per le implementazioni di produzione perché è in linea con le AWS migliori pratiche per le implement azioni di configurazione.

Strategia di distribuzione	Descrizione
AppConfig.AllAtOnce	Rapidità: questa strategia distribuisce immediatamente la configurazione a tutte le destinazioni. Il sistema monitora gli allarmi per 10 minuti. CloudWatch Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completat a. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione.
AppConfig. Lineare 50 30 secondi PercentEv ery	Test/dimostrazione: questa strategia distribuisce la configurazione a metà di tutte le destinazioni ogni 30 secondi per una distribuzione di un minuto. Il sistema monitora gli CloudWatch allarmi Amazon per 1 minuto. Se non vengono ricevuti allarmi in questo periodo, la distribuzione è completata. Se durante questo periodo viene attivato un avviso, eseguire il rollback AWS AppConfig della distribuzione. Ti consigliamo di utilizzare questa strategia solo a scopo di test o dimostrazione perché ha una breve durata e tempo di bake.

Creazione di una strategia di distribuzione

È possibile creare un massimo di 20 strategie di distribuzione. Quando si distribuisce una configurazione, è possibile scegliere la strategia di distribuzione più adatta all'applicazione e all'ambiente.

Creazione di una strategia AWS AppConfig di implementazione (console)

Utilizzare la procedura seguente per creare una strategia di distribuzione AWS AppConfig utilizzando la console AWS Systems Manager.

Per creare una strategia di distribuzione

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel pannello di navigazione, scegliere AWS AppConfig.
3. Scegliere la scheda Deployment Strategies (Strategie di distribuzione) e quindi scegliere Create deployment strategy (Crea strategia di distribuzione).
4. In Name (Nome), immettere un nome per la strategia di distribuzione.
5. In Description (Descrizione), immettere informazioni sulla strategia di distribuzione.
6. In Tipo di distribuzione, scegliere un tipo.
7. In Step percentage (Percentuale fase), scegliere la percentuale di chiamanti da destinare durante ogni passaggio della distribuzione.
8. In Deployment time (Tempo di distribuzione), immettere la durata totale della distribuzione espressa in minuti o ore.
9. Per Bake time, inserisci il tempo totale, in minuti o ore, per monitorare gli CloudWatch allarmi Amazon prima di procedere alla fase successiva di una distribuzione o prima di considerarla completa.
10. Nella sezione Tags (Tag) immettere una chiave e un valore facoltativo. È possibile specificare un massimo di 50 tag per una risorsa.
11. Scegliere Create deployment strategy (Crea strategia di distribuzione).

> **Important**
>
> Se hai creato un profilo di configurazione perAWS CodePipeline, devi creare una pipeline CodePipeline che specifichi AWS AppConfig come provider di distribuzione. Non è necessario eseguire. Distribuzione di una configurazione Tuttavia, è necessario configurare un client per ricevere gli aggiornamenti della configurazione dell'applicazione come descritto inRecupero delle configurazioni chiamando direttamente le API. Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di

distribuzione, vedi Tutorial: Create a Pipeline that Uses AWS AppConfig as a Deployment Provider nella Guida per l'utente. AWS CodePipeline

Continua con la Distribuzione di una configurazione.

Creazione di una strategia di AWS AppConfig distribuzione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Tools for PowerShell creare una strategia di AWS AppConfig distribuzione.

Per creare una strategia di distribuzione passo dopo passo

1. Aprire AWS CLI.
2. Esegui il comando seguente per creare una strategia di distribuzione.

   Linux

```
aws appconfig create-deployment-strategy \
  --name A_name_for_the_deployment_strategy \
  --description A_description_of_the_deployment_strategy \
  --deployment-duration-in-minutes Total_amount_of_time_for_a_deployment_to_last
 \
  --final-bake-time-in-minutes Amount_of_time_AWS
 AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
 \
  --growth-
factor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_interva
 \
  --growth-
type The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_time
 \
  --replicate-
to To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document \
  --tags User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

   Windows

```
aws appconfig create-deployment-strategy ^
  --name A_name_for_the_deployment_strategy ^
  --description A_description_of_the_deployment_strategy ^
```

```
  --deployment-duration-in-minutes Total_amount_of_time_for_a_deployment_to_last
 ^
  --final-bake-time-in-minutes Amount_of_time_AWS
 AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
 ^
  --growth-
factor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_interva
 ^
  --growth-
type The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_time
 ^
  --name A_name_for_the_deployment_strategy ^
  --replicate-
to To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document ^
  --tags User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

PowerShell

```
New-APPCDeploymentStrategy `
  --Name A_name_for_the_deployment_strategy `
  --Description A_description_of_the_deployment_strategy `
  --DeploymentDurationInMinutes Total_amount_of_time_for_a_deployment_to_last `
  --FinalBakeTimeInMinutes Amount_of_time_AWS
 AppConfig_monitors_for_alarms_before_considering_the_deployment_to_be_complete
 `
  --
GrowthFactor The_percentage_of_targets_to_receive_a_deployed_configuration_during_each_i
 `
  --
GrowthType The_linear_or_exponential_algorithm_used_to_define_how_percentage_grows_over_
 `
  --
ReplicateTo To_save_the_deployment_strategy_to_a_Systems_Manager_(SSM)_document
 `
  --
Tag Hashtable_type_User_defined_key_value_pair_metadata_of_the_deployment_strategy
```

Il sistema restituisce informazioni simili alle seguenti.

Linux

```
{
```

```
   "Id": "Id of the deployment strategy",
   "Name": "Name of the deployment strategy",
   "Description": "Description of the deployment strategy",
   "DeploymentDurationInMinutes": "Total amount of time the deployment lasted",
   "GrowthType": "The linear or exponential algorithm used to define how
 percentage grew over time",
   "GrowthFactor": "The percentage of targets that received a deployed
 configuration during each interval",
   "FinalBakeTimeInMinutes": "The amount of time AWS AppConfig monitored for
 alarms before considering the deployment to be complete",
   "ReplicateTo": "The Systems Manager (SSM) document where the deployment
 strategy is saved"
}
```

Windows

```
{
   "Id": "Id of the deployment strategy",
   "Name": "Name of the deployment strategy",
   "Description": "Description of the deployment strategy",
   "DeploymentDurationInMinutes": "Total amount of time the deployment lasted",
   "GrowthType": "The linear or exponential algorithm used to define how
 percentage grew over time",
   "GrowthFactor": "The percentage of targets that received a deployed
 configuration during each interval",
   "FinalBakeTimeInMinutes": "The amount of time AWS AppConfig monitored for
 alarms before considering the deployment to be complete",
   "ReplicateTo": "The Systems Manager (SSM) document where the deployment
 strategy is saved"
}
```

PowerShell

```
ContentLength               : Runtime of the command
DeploymentDurationInMinutes : Total amount of time the deployment lasted
Description                 : Description of the deployment strategy
FinalBakeTimeInMinutes      : The amount of time AWS AppConfig monitored for
 alarms before considering the deployment to be complete
GrowthFactor                : The percentage of targets that received a deployed
 configuration during each interval
GrowthType                  : The linear or exponential algorithm used to define
 how percentage grew over time
HttpStatusCode              : HTTP Status of the runtime
```

```
Id                          : The deployment strategy ID
Name                        : Name of the deployment strategy
ReplicateTo                 : The Systems Manager (SSM) document where the
 deployment strategy is saved
ResponseMetadata            : Runtime Metadata
```

Distribuzione di una configurazione

Dopo aver creato gli elementi necessari per lavorare con i flag di funzionalità e i dati di configurazione in formato libero, puoi creare una nuova distribuzione utilizzando l'AWS Management Console, l'o l'SDKAWS CLI. L'avvio di una distribuzione richiede l'operazione dell'APIAWS AppConfig. StartDeployment Questa chiamata include gli ID dell'applicazione, dell'ambiente, del profilo di configurazione e (facoltativamente) della versione dei dati di configurazione AWS AppConfig da distribuire. La chiamata include anche l'ID della strategia di distribuzione da utilizzare, che determina la modalità di distribuzione dei dati di configurazione.

Se distribuisci segreti archiviati in AWS Secrets Manager oggetti Amazon Simple Storage Service (Amazon S3) crittografati con una chiave gestita dal cliente o parametri di stringa sicuri archiviati in AWS Systems Manager Parameter Store crittografati con una chiave gestita dal cliente, devi specificare un valore per il parametro. `KmsKeyIdentifier` Se la configurazione non è crittografata o è crittografata con unChiave gestita da AWS, non è necessario specificare un valore per il `KmsKeyIdentifier` parametro.

> **Note**
>
> Il valore specificato `KmsKeyIdentifier` deve essere una chiave gestita dal cliente. Non deve essere necessariamente la stessa chiave che hai usato per crittografare la configurazione.
> Quando si avvia una distribuzione con un`KmsKeyIdentifier`, la politica di autorizzazione allegata al principale AWS Identity and Access Management (IAM) deve consentire l'operazione. `kms:GenerateDataKey`

AWS AppConfig monitora la distribuzione su tutti gli host e lo stato dei report. Se una distribuzione non riesce, AWS AppConfig esegue il rollback della configurazione.

> **Note**
>
> È possibile implementare solo una configurazione alla volta in un ambiente. Tuttavia, è possibile implementare una configurazione ciascuna in ambienti diversi contemporaneamente.

Implementa una configurazione (console)

Utilizzare la procedura seguente per distribuire una configurazione AWS AppConfig mediante la console AWS Systems Manager.

Per distribuire una configurazione utilizzando la console

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel pannello di navigazione, scegliere AWS AppConfig.
3. Nella scheda Applications (Applicazioni), scegliere un'applicazione, quindi scegliere View details (Visualizza dettagli).
4. Nella scheda Environments (Ambienti), scegliere un ambiente e quindi scegliere View details (Visualizza dettagli).
5. Selezionare Start deployment (Avvia distribuzione).
6. Per Configuration (Configurazione), scegliere una configurazione dall'elenco.
7. A seconda dell'origine della configurazione, utilizzare l'elenco Document version (Versione documento) o Parameter version (Versione Parametro) per scegliere la versione che si desidera distribuire.
8. In Deployment strategy (Strategia di distribuzione), scegliere una strategia dall'elenco.
9. In Deployment description (Descrizione distribuzione), immettere una descrizione.
10. Nella sezione Tags (Tag) immettere una chiave e un valore facoltativo. È possibile specificare un massimo di 50 tag per una risorsa.
11. Selezionare Start deployment (Avvia distribuzione).

Implementa una configurazione (riga di comando)

La procedura seguente descrive come utilizzare AWS CLI (su Linux o Windows) o AWS Tools for PowerShell distribuire una configurazione. AWS AppConfig

Per distribuire una configurazione passo dopo passo

1. Aprire AWS CLI.
2. Esegui il comando seguente per distribuire una configurazione.

   Linux

   ```
   aws appconfig start-deployment \
     --application-id The_application_ID \
     --environment-id The_environment_ID \
     --deployment-strategy-id The_deployment_strategy_ID \
     --configuration-profile-id The_configuration_profile_ID \
     --configuration-version The_configuration_version_to_deploy \
     --description A_description_of_the_deployment \
     --tags User_defined_key_value_pair_metadata_of_the_deployment
   ```

   Windows

   ```
   aws appconfig start-deployment ^
     --application-id The_application_ID ^
     --environment-id The_environment_ID ^
     --deployment-strategy-id The_deployment_strategy_ID ^
     --configuration-profile-id The_configuration_profile_ID ^
     --configuration-version The_configuration_version_to_deploy ^
     --description A_description_of_the_deployment ^
     --tags User_defined_key_value_pair_metadata_of_the_deployment
   ```

   PowerShell

   ```
   Start-APPCDeployment `
     -ApplicationId The_application_ID `
     -ConfigurationProfileId The_configuration_profile_ID `
     -ConfigurationVersion The_configuration_version_to_deploy `
     -DeploymentStrategyId The_deployment_strategy_ID `
     -Description A_description_of_the_deployment `
     -EnvironmentId The_environment_ID `
   ```

```
  -Tag Hashtable_type_user_defined_key_value_pair_metadata_of_the_deployment
```

Il sistema restituisce informazioni simili alle seguenti.

Linux

```
{
   "ApplicationId": "The ID of the application that was deployed",
   "EnvironmentId" : "The ID of the environment",
   "DeploymentStrategyId": "The ID of the deployment strategy that was
 deployed",
   "ConfigurationProfileId": "The ID of the configuration profile that was
 deployed",
   "DeploymentNumber": The sequence number of the deployment,
   "ConfigurationName": "The name of the configuration",
   "ConfigurationLocationUri": "Information about the source location of the
 configuration",
   "ConfigurationVersion": "The configuration version that was deployed",
   "Description": "The description of the deployment",
   "DeploymentDurationInMinutes": Total amount of time the deployment lasted,
   "GrowthType": "The linear or exponential algorithm used to define how
 percentage grew over time",
   "GrowthFactor": The percentage of targets to receive a deployed configuration
 during each interval,
   "FinalBakeTimeInMinutes": Time AWS AppConfig monitored for alarms before
 considering the deployment to be complete,
   "State": "The state of the deployment",

   "EventLog": [
      {
         "Description": "A description of the deployment event",
         "EventType": "The type of deployment event",
         "OccurredAt": The date and time the event occurred,
         "TriggeredBy": "The entity that triggered the deployment event"
      }
   ],

   "PercentageComplete": The percentage of targets for which the deployment is
 available,
   "StartedAt": The time the deployment started,
   "CompletedAt": The time the deployment completed
}
```

Windows

```
{
   "ApplicationId": "The ID of the application that was deployed",
   "EnvironmentId" : "The ID of the environment",
   "DeploymentStrategyId": "The ID of the deployment strategy that was
 deployed",
   "ConfigurationProfileId": "The ID of the configuration profile that was
 deployed",
   "DeploymentNumber": The sequence number of the deployment,
   "ConfigurationName": "The name of the configuration",
   "ConfigurationLocationUri": "Information about the source location of the
 configuration",
   "ConfigurationVersion": "The configuration version that was deployed",
   "Description": "The description of the deployment",
   "DeploymentDurationInMinutes": Total amount of time the deployment lasted,
   "GrowthType": "The linear or exponential algorithm used to define how
 percentage grew over time",
   "GrowthFactor": The percentage of targets to receive a deployed configuration
 during each interval,
   "FinalBakeTimeInMinutes": Time AWS AppConfig monitored for alarms before
 considering the deployment to be complete,
   "State": "The state of the deployment",

   "EventLog": [
      {
         "Description": "A description of the deployment event",
         "EventType": "The type of deployment event",
         "OccurredAt": The date and time the event occurred,
         "TriggeredBy": "The entity that triggered the deployment event"
      }
   ],

   "PercentageComplete": The percentage of targets for which the deployment is
 available,
   "StartedAt": The time the deployment started,
   "CompletedAt": The time the deployment completed
}
```

PowerShell

```
ApplicationId               : The ID of the application that was deployed
```

```
CompletedAt                  : The time the deployment completed
ConfigurationLocationUri     : Information about the source location of the
 configuration
ConfigurationName            : The name of the configuration
ConfigurationProfileId       : The ID of the configuration profile that was
 deployed
ConfigurationVersion         : The configuration version that was deployed
ContentLength                : Runtime of the deployment
DeploymentDurationInMinutes : Total amount of time the deployment lasted
DeploymentNumber             : The sequence number of the deployment
DeploymentStrategyId         : The ID of the deployment strategy that was
 deployed
Description                  : The description of the deployment
EnvironmentId                : The ID of the environment that was deployed
EventLog                     : {Description : A description of the deployment
 event, EventType : The type of deployment event, OccurredAt : The date and time
 the event occurred,
         TriggeredBy : The entity that triggered the deployment event}
FinalBakeTimeInMinutes       : Time AWS AppConfig monitored for alarms before
 considering the deployment to be complete
GrowthFactor                 : The percentage of targets to receive a deployed
 configuration during each interval
GrowthType                   : The linear or exponential algorithm used to define
 how percentage grew over time
HttpStatusCode               : HTTP Status of the runtime
PercentageComplete           : The percentage of targets for which the deployment
 is available
ResponseMetadata             : Runtime Metadata
StartedAt                    : The time the deployment started
State                        : The state of the deployment
```

AWS AppConfigimplementazione, integrazione con CodePipeline

AWS AppConfigè un'azione di implementazione integrata per AWS CodePipeline (CodePipeline). CodePipeline è un servizio di distribuzione continua completamente gestito che consente di automatizzare le pipeline di rilascio per aggiornamenti rapidi e affidabili di applicazioni e infrastrutture. CodePipeline automatizza le fasi di compilazione, test e distribuzione del processo di rilascio ogni volta che viene apportata una modifica al codice, in base al modello di rilascio definito dall'utente. Per ulteriori informazioni, consulta Che cos'è AWS CodePipeline?

L'integrazione di AWS AppConfig with CodePipeline offre i seguenti vantaggi:

- I clienti che CodePipeline gestivano l'orchestrazione ora dispongono di un mezzo leggero per implementare le modifiche alla configurazione delle proprie applicazioni senza dover implementare l'intera base di codice.
- I clienti che desiderano AWS AppConfig gestire le implementazioni di configurazione ma sono limitati perché AWS AppConfig non supportano il codice o l'archivio di configurazione correnti, ora dispongono di opzioni aggiuntive. CodePipeline supporta AWS CodeCommit GitHub, e BitBucket (solo per citarne alcuni).

> **Note**
>
> AWS AppConfigl'integrazione con CodePipeline è supportata solo Regioni AWS dove CodePipeline è disponibile.

Come funziona l'integrazione

Si inizia con l'impostazione e la configurazione CodePipeline. Ciò include l'aggiunta della configurazione a un archivio CodePipeline di codice supportato. Successivamente, configuri il tuo AWS AppConfig ambiente eseguendo le seguenti attività:

- Crea uno spazio dei nomi e un profilo di configurazione
- Scegli una strategia di implementazione predefinita o creane una personalizzata

Dopo aver completato queste attività, crei una pipeline CodePipeline che specifica AWS AppConfig come provider di distribuzione. Puoi quindi apportare una modifica alla configurazione e caricarla nel tuo CodePipeline code store. Il caricamento della nuova configurazione avvia automaticamente una nuova distribuzione in CodePipeline. Una volta completata la distribuzione, puoi verificare le modifiche. Per informazioni sulla creazione di una pipeline che viene specificata AWS AppConfig come provider di distribuzione, vedi Tutorial: Create a Pipeline That Uses AWS AppConfig as a Deployment Provider nella Guida per l'utente. AWS CodePipeline

Recupero dei flag delle funzionalità e dei dati di configurazione in AWS AppConfig

L'applicazione recupera i flag delle funzionalità e i dati di configurazione in formato libero stabilendo una sessione di configurazione utilizzando il servizio Data. AWS AppConfig Se si utilizza uno dei metodi di recupero semplificati descritti in questa sezione, l'estensione Agent AWS AppConfig Lambda o AWS AppConfig Agent gestisce una serie di chiamate API e token di sessione per conto dell'utente. AWS AppConfig L'agente viene configurato come host locale e l'agente effettua il sondaggio per gli aggiornamenti della configurazione. AWS AppConfig L'agente richiama le azioni StartConfigurationSessione GetLatestConfigurationAPI e memorizza nella cache i dati di configurazione localmente. Per recuperare i dati, l'applicazione effettua una chiamata HTTP al server localhost. AWS AppConfig L'agente supporta diversi casi d'uso, come descritto in. Metodi di recupero semplificati

Se preferisci, puoi richiamare manualmente queste azioni API per recuperare una configurazione. Il processo API funziona come segue:

L'applicazione stabilisce una sessione di configurazione utilizzando l'azione `StartConfigurationSession` API. Il client della sessione effettua quindi chiamate periodiche per `GetLatestConfiguration` verificare e recuperare i dati più recenti disponibili.

Durante la chiamata`StartConfigurationSession`, il codice invia gli identificatori (ID o nome) di un' AWS AppConfig applicazione, di un ambiente e di un profilo di configurazione monitorati dalla sessione.

In risposta, AWS AppConfig fornisce un `InitialConfigurationToken` codice da fornire al client della sessione e da utilizzare la prima volta che richiama `GetLatestConfiguration` quella sessione.

Durante la chiamata`GetLatestConfiguration`, il codice client invia il `ConfigurationToken` valore più recente a sua disposizione e riceve in risposta:

- `NextPollConfigurationToken`: il `ConfigurationToken` valore da utilizzare nella chiamata successiva a`GetLatestConfiguration`.
- La configurazione: i dati più recenti destinati alla sessione. Questo campo può essere vuoto se il client dispone già dell'ultima versione della configurazione.

Questa sezione include le seguenti informazioni.

Indice

Informazioni sul servizio AWS AppConfig Data Plane

Il 18 novembre 2021, AWS AppConfig ha rilasciato un nuovo servizio di piano dati. Questo servizio sostituisce il precedente processo di recupero dei dati di configurazione utilizzando l'azione API. `GetConfiguration` Il servizio data plane utilizza due nuove azioni API e. StartConfigurationSessionGetLatestConfiguration Il servizio data plane utilizza anche nuovi endpoint.

Se hai iniziato a utilizzarla AWS AppConfig prima del 28 gennaio 2022, il servizio potrebbe richiamare direttamente l'azione `GetConfiguration` API o potrebbe utilizzare un client fornito da AWS, come l'estensione AWS AppConfig Agent Lambda, per richiamare questa azione API. Se richiami direttamente l'azione `GetConfiguration` API, prendi provvedimenti per utilizzare le azioni `StartConfigurationSession` e `GetLatestConfiguration` API. Se si utilizza l'estensione AWS AppConfig Agent Lambda, vedere la sezione intitolata Come questa modifica influisce sull'estensione Agent AWS AppConfig Lambda più avanti in questo argomento.

Le nuove azioni API del piano dati offrono i seguenti vantaggi rispetto all'azione `GetConfiguration` API, che ora è obsoleta.

1. Non è necessario gestire un parametro. `ClientID` Con il servizio data plane, `ClientID` viene gestito internamente dal token di sessione creato da`StartConfigurationSession`.
2. Non è più necessario includere o `ClientConfigurationVersion` indicare la versione memorizzata nella cache dei dati di configurazione. Con il servizio data plane, `ClientConfigurationVersion` è gestito internamente dal token di sessione creato da. `StartConfigurationSession`
3. Il nuovo endpoint dedicato per le chiamate API del piano dati migliora la struttura del codice separando le chiamate al piano di controllo da quelle sul piano dati.
4. Il nuovo servizio data plane migliora l'estensibilità futura per le operazioni del piano dati. Utilizzando una sessione di configurazione che gestisce il recupero dei dati di configurazione, il AWS AppConfig team può creare miglioramenti più potenti in futuro.

Migrazione da **`GetConfiguration`** a **`GetLatestConfiguration`**

Per iniziare a utilizzare il nuovo servizio Data Plane, è necessario aggiornare il codice che richiama l'azione dell'API. `GetConfiguration` Avvia una sessione di configurazione utilizzando l'azione `StartConfigurationSession` API, quindi richiama l'azione `GetLatestConfiguration` API per recuperare i dati di configurazione. Per migliorare le prestazioni, ti consigliamo di memorizzare nella cache i dati di configurazione localmente. Per ulteriori informazioni, consulta Recupero delle configurazioni chiamando direttamente le API.

In che modo questa modifica influisce sull'estensione AWS AppConfig Agent Lambda

Questa modifica non ha alcun impatto diretto sul funzionamento dell'estensione AWS AppConfig Agent Lambda. Le versioni precedenti dell'estensione AWS AppConfig Agent Lambda chiamavano l'azione `GetConfiguration` API per tuo conto. Le versioni più recenti chiamano azioni API del piano dati. Se utilizzi l'estensione AWS AppConfig Lambda, ti consigliamo di aggiornare l'estensione all'Amazon Resource Name (ARN) più recente e di aggiornare le autorizzazioni per le nuove chiamate API. Per ulteriori informazioni, consulta Recupero dei dati di configurazione utilizzando l'estensione Agent AWS AppConfig Lambda.

Metodi di recupero semplificati

AWS AppConfig offre diversi metodi semplificati per il recupero dei dati di configurazione. Se si utilizzano flag di AWS AppConfig funzionalità o dati di configurazione in formato libero in una AWS Lambda funzione, è possibile utilizzare l'estensione AWS AppConfig Agent Lambda per recuperare le configurazioni. Se hai applicazioni in esecuzione su istanze Amazon EC2, puoi utilizzare AWS AppConfig Agent per recuperare le configurazioni. AWS AppConfig Agent supporta anche applicazioni in esecuzione su immagini di container Amazon Elastic Kubernetes Service (Amazon EKS) o Amazon Elastic Container Service (Amazon ECS).

Dopo aver completato la configurazione iniziale, questi metodi di recupero dei dati di configurazione sono più semplici rispetto alla chiamata diretta delle API. AWS AppConfig Implementano automaticamente le migliori pratiche e possono ridurre i costi di utilizzo grazie AWS AppConfig al minor numero di chiamate API per il recupero delle configurazioni.

Argomenti

- Recupero dei dati di configurazione utilizzando l'estensione Agent AWS AppConfig Lambda
- Recupero dei dati di configurazione dalle istanze Amazon EC2

- Recupero dei dati di configurazione da Amazon ECS e Amazon EKS
- Funzionalità di recupero aggiuntive
- AWS AppConfig Sviluppo locale dell'agente

Recupero dei dati di configurazione utilizzando l'estensione Agent AWS AppConfig Lambda

Un' AWS Lambda estensione è un processo complementare che aumenta le funzionalità di una funzione Lambda. Un'estensione può iniziare prima che una funzione venga richiamata, essere eseguita in parallelo con una funzione e continuare a funzionare dopo l'elaborazione della chiamata di una funzione. In sostanza, un'estensione Lambda è come un client che viene eseguito in parallelo a una chiamata Lambda. Questo client parallelo può interfacciarsi con la funzione in qualsiasi momento del suo ciclo di vita.

Se utilizzi i flag di AWS AppConfig funzionalità o altri dati di configurazione dinamici in una funzione Lambda, ti consigliamo di aggiungere l'estensione AWS AppConfig Agent Lambda come livello alla tua funzione Lambda. Ciò semplifica la chiamata ai flag delle funzionalità e l'estensione stessa include le migliori pratiche che ne semplificano l'utilizzo riducendo al contempo i costi. AWS AppConfig I costi ridotti derivano da un minor numero di chiamate API al AWS AppConfig servizio e da tempi di elaborazione delle funzioni Lambda più brevi. Per ulteriori informazioni sulle estensioni Lambda, consulta le estensioni Lambda nella Developer Guide.AWS Lambda

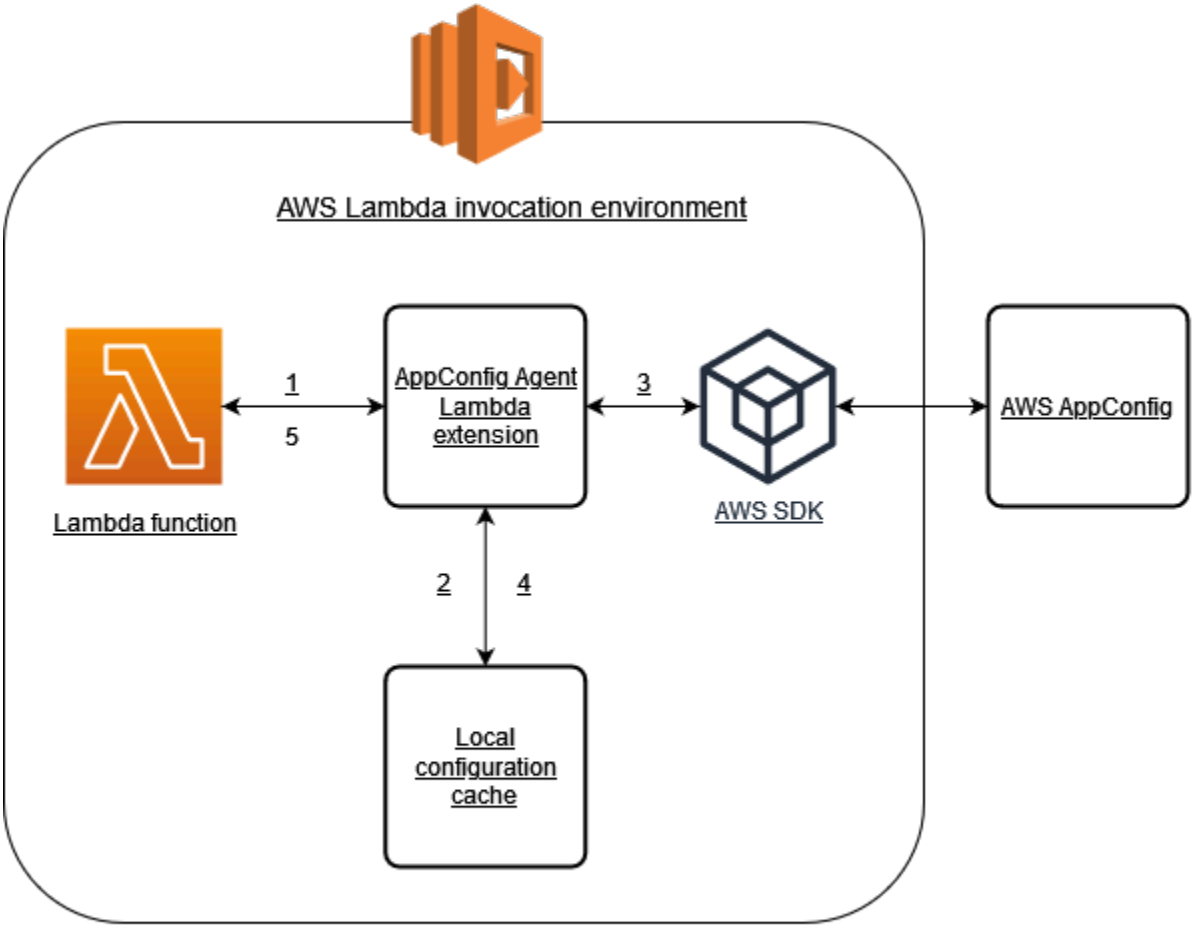
> **Note**
>
> AWS AppConfig i prezzi si basano sul numero di volte in cui una configurazione viene chiamata e ricevuta. I costi aumentano se la Lambda esegue più partenze a freddo e recupera frequentemente nuovi dati di configurazione.

Questo argomento include informazioni sull'estensione AWS AppConfig Agent Lambda e la procedura per configurare l'estensione in modo che funzioni con la funzione Lambda.

Come funziona

Se utilizzi AWS AppConfig per gestire le configurazioni per una funzione Lambda senza estensioni Lambda, devi configurare la funzione Lambda per ricevere aggiornamenti di configurazione mediante l'integrazione con le azioni e API. StartConfigurationSessionGetLatestConfiguration

L'integrazione dell'estensione AWS AppConfig Agent Lambda con la funzione Lambda semplifica questo processo. L'estensione si occupa della chiamata al AWS AppConfig servizio, della gestione di una cache locale dei dati recuperati, del tracciamento dei token di configurazione necessari per le successive chiamate di servizio e del controllo periodico degli aggiornamenti di configurazione in background. Il diagramma seguente mostra come funziona.



1. L'estensione AWS AppConfig Agent Lambda viene configurata come livello della funzione Lambda.
2. Per accedere ai dati di configurazione, la funzione chiama l' AWS AppConfig estensione su un endpoint HTTP in esecuzione. `localhost:2772`
3. L'estensione mantiene una cache locale dei dati di configurazione. Se i dati non sono nella cache, l'estensione chiama AWS AppConfig per ottenere i dati di configurazione.
4. Dopo aver ricevuto la configurazione dal servizio, l'estensione la memorizza nella cache locale e la passa alla funzione Lambda.

5. AWS AppConfig L'estensione Agent Lambda verifica periodicamente la presenza di aggiornamenti dei dati di configurazione in background. Ogni volta che viene richiamata la funzione Lambda, l'estensione controlla il tempo trascorso da quando ha recuperato una configurazione. Se il tempo trascorso è maggiore dell'intervallo di sondaggio configurato, l'estensione chiama AWS AppConfig per verificare la presenza di nuovi dati distribuiti, aggiorna la cache locale in caso di modifiche e reimposta il tempo trascorso.

> **Note**
>
> - Lambda crea istanze separate corrispondenti al livello di simultaneità richiesto dalla funzione. Ogni istanza è isolata e mantiene la propria cache locale dei dati di configurazione. Per ulteriori informazioni sulle istanze Lambda e sulla concorrenza, vedere Gestione della concorrenza per una funzione Lambda.
> - Il tempo necessario affinché una modifica alla configurazione appaia in una funzione Lambda, dopo aver distribuito una configurazione aggiornata da AWS AppConfig, dipende dalla strategia di distribuzione utilizzata per la distribuzione e dall'intervallo di polling configurato per l'estensione.

Prima di iniziare

Prima di abilitare l'estensione AWS AppConfig Agent Lambda, procedi come segue:

- Organizza le configurazioni nella tua funzione Lambda in modo da poterle esternalizzare. AWS AppConfig
- Crea AWS AppConfig artefatti e dati di configurazione, inclusi contrassegni di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig.
- Aggiungi `appconfig:StartConfigurationSession` e `appconfig:GetLatestConfiguration` alla policy AWS Identity and Access Management (IAM) utilizzata dal ruolo di esecuzione della funzione Lambda. Per ulteriori informazioni, consulta Ruolo di esecuzione di AWS Lambda nella Guida per gli sviluppatori di AWS Lambda . Per ulteriori informazioni sulle AWS AppConfig autorizzazioni, consulta Azioni, risorse e chiavi di condizione AWS AppConfig nel Service Authorization Reference.

Aggiungere l'estensione AWS AppConfig Agent Lambda

Per utilizzare l'estensione AWS AppConfig Agent Lambda, devi aggiungere l'estensione alla tua Lambda. Questo può essere fatto aggiungendo l'estensione AWS AppConfig Agent Lambda alla funzione Lambda come livello o abilitando l'estensione su una funzione Lambda come immagine contenitore.

> **Note**
>
> L' AWS AppConfig estensione è indipendente dal runtime e supporta tutti i runtime.

Aggiungere l'estensione AWS AppConfig Agent Lambda utilizzando un layer e un ARN

Per utilizzare l'estensione AWS AppConfig Agent Lambda, aggiungete l'estensione alla funzione Lambda come livello. Per informazioni su come aggiungere un livello alla funzione, consulta Configuring extensions nella Developer Guide.AWS Lambda Il nome dell'estensione nella AWS Lambda console è AWS- AppConfig -Extension. Tieni inoltre presente che quando aggiungi l'estensione come layer a Lambda, devi specificare un Amazon Resource Name (ARN). Scegli un ARN da uno dei seguenti elenchi che corrisponde alla piattaforma e Regione AWS dove hai creato la Lambda.

- piattaforma x86-64
- Piattaforma ARM64

Se desideri testare l'estensione prima di aggiungerla alla tua funzione, puoi verificarne il funzionamento utilizzando il seguente esempio di codice.

```
import urllib.request


def lambda_handler(event, context):
    url = f'http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name'
    config = urllib.request.urlopen(url).read()
    return config
```

Per testarlo, crea una nuova funzione Lambda per Python, aggiungi l'estensione e quindi esegui la funzione Lambda. Dopo aver eseguito la funzione Lambda, la funzione AWS AppConfig Lambda

restituisce la configurazione specificata per il percorso http://localhost:2772. Per informazioni sulla creazione di una funzione Lambda, consulta Create a Lambda function with the console nella Developer Guide.AWS Lambda

Per aggiungere l'estensione AWS AppConfig Agent Lambda come immagine del contenitore, vedere. Utilizzo di un'immagine del contenitore per aggiungere l'estensione AWS AppConfig Agent Lambda

Configurazione dell'estensione AWS AppConfig Agent Lambda

È possibile configurare l'estensione modificando le seguenti variabili di AWS Lambda ambiente. Per ulteriori informazioni, consulta Uso delle variabili di AWS Lambda ambiente nella Guida per gli AWS Lambda sviluppatori.

Preacquisizione dei dati di configurazione

La variabile di ambiente `AWS_APPCONFIG_EXTENSION_PREFETCH_LIST` può migliorare il tempo di avvio della funzione. Quando l'estensione AWS AppConfig Agent Lambda viene inizializzata, recupera la configurazione specificata da prima che AWS AppConfig Lambda inizi a inizializzare la funzione e a richiamare il gestore. In alcuni casi, i dati di configurazione sono già disponibili nella cache locale prima che la funzione li richieda.

Per utilizzare la funzionalità di prefetch, impostate il valore della variabile di ambiente sul percorso corrispondente ai dati di configurazione. Ad esempio, se la configurazione corrisponde a un'applicazione, un ambiente e un profilo di configurazione denominati rispettivamente «my_application», «my_environment» e «my_configuration_data», il percorso sarebbe. `/applications/my_application/environments/my_environment/configurations/my_configuration_data` Puoi specificare più elementi di configurazione elencandoli come elenco separato da virgole (se hai un nome di risorsa che include una virgola, usa il valore ID della risorsa anziché il suo nome).

Accesso ai dati di configurazione da un altro account

L'estensione AWS AppConfig Agent Lambda può recuperare i dati di configurazione da un altro account specificando un ruolo IAM che concede le autorizzazioni ai dati. Per configurarlo, segui questi passaggi:

1. Nell'account in cui AWS AppConfig viene utilizzato per gestire i dati di configurazione, crea un ruolo con una policy di fiducia che conceda all'account che esegue la funzione Lambda l'accesso alle `appconfig:StartConfigurationSession` azioni

`appconfig:GetLatestConfiguration` and, insieme agli ARN parziali o completi corrispondenti alle AWS AppConfig risorse di configurazione.

2. Nell'account che esegue la funzione Lambda, aggiungi la variabile di `AWS_APPCONFIG_EXTENSION_ROLE_ARN` ambiente alla funzione Lambda con l'ARN del ruolo creato nel passaggio 1.
3. (Facoltativo) Se necessario, è possibile specificare un ID esterno utilizzando la `AWS_APPCONFIG_EXTENSION_ROLE_EXTERNAL_ID` variabile di ambiente. Allo stesso modo, è possibile configurare un nome di sessione utilizzando la variabile di `AWS_APPCONFIG_EXTENSION_ROLE_SESSION_NAME` ambiente.

> **Note**
>
> Osservare le seguenti informazioni.
>
> - L'estensione AWS AppConfig Agent Lambda può recuperare i dati da un solo account. Se specifichi un ruolo IAM, l'estensione non sarà in grado di recuperare i dati di configurazione dall'account in cui è in esecuzione la funzione Lambda.
> - AWS Lambda registra le informazioni sull'estensione AWS AppConfig Agent Lambda e sulla funzione Lambda utilizzando Amazon Logs. CloudWatch

Variabile di ambiente	Informazioni	Valore predefinito
`AWS_APPCONFIG_EXTENSION_HTTP_PORT`	Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP locale che ospita l'estensione.	2772
`AWS_APPCONFIG_EXTENSION_LOG_LEVEL`	Questa variabile di ambiente specifica quali log AWS AppConfig specifici dell'estensione vengono inviati ad Amazon CloudWatch Logs per una funzione. I valori validi senza distinzione tra maiuscole e minuscole sono:,,,	`info`

Variabile di ambiente	Informazioni	Valore predefinito
	e. `debug` `info` `warn` `error` `none` Debug include informazioni dettagliate, incluse informazioni sulla tempistica, sull'estensione.	
`AWS_APPCONFIG_EXTENSION_MAX_CONNECTIONS`	Questa variabile di ambiente configura il numero massimo di connessioni da cui l'estensione utilizza per recuperare le configurazioni. AWS AppConfig	3
`AWS_APPCONFIG_EXTENSION_POLL_INTERVAL_SECONDS`	Questa variabile di ambiente controlla la frequenza con cui l'estensione richiede una configurazione AWS AppConfig aggiornata in secondi.	45
`AWS_APPCONFIG_EXTENSION_POLL_TIMEOUT_MILLIS`	Questa variabile di ambiente controlla il tempo massimo, in millisecondi, in cui l'estensione attende una risposta AWS AppConfig durante l'aggiornamento dei dati nella cache. Se AWS AppConfig non risponde nel periodo di tempo specificato, l'estensione salta questo intervallo di sondaggio e restituisce i dati memorizzati nella cache precedentemente aggiornati.	3000

Variabile di ambiente	Informazioni	Valore predefinito
`AWS_APPCONFIG_EXTENSION_PREFETCH_LIST`	Questa variabile di ambiente specifica i dati di configurazione che l'estensione inizia a recuperare prima che la funzione venga inizializzata e il gestore venga eseguito. Può ridurre in modo significativo il tempo di avvio a freddo della funzione.	Nessuno
`AWS_APPCONFIG_EXTENSION_PROXY_HEADERS`	Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente. `AWS_APPCONFIG_EXTENSION_PROXY_URL` Il valore è un elenco di intestazioni separate da virgole. Ogni intestazione utilizza il seguente modulo: `"header: value"`	Nessuno
`AWS_APPCONFIG_EXTENSION_PROXY_URL`	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall' AWS AppConfig estensione a. Servizi AWSHTTPSe gli `HTTP` URL sono supportati.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
`AWS_APPCONFIG_EXTENSION_ROLE_ARN`	Questa variabile di ambiente specifica l'ARN del ruolo IAM corrispondente a un ruolo che deve essere assunto dall'estensione per recuperare AWS AppConfig la configurazione.	Nessuno
`AWS_APPCONFIG_EXTENSION_ROLE_EXTERNAL_ID`	Questa variabile di ambiente specifica l'id esterno da utilizzare insieme al ruolo ARN assunto.	Nessuno
`AWS_APPCONFIG_EXTENSION_ROLE_SESSION_NAME`	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno
`AWS_APPCONFIG_EXTENSION_SERVICE_REGION`	Questa variabile di ambiente specifica una regione alternativa che l'estensione deve utilizzare per chiamare il servizio. AWS AppConfig Se non è definita, l'estensione utilizza l'endpoint nella regione corrente.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
`AWS_APPCONFIG_EXTENSION_MANIFEST`	Questa variabile di ambiente configura l' AWS AppConfig agente per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. È possibile inserire uno dei seguenti valori: • `"app:env:manifest-config"` • `"file:/fully/qualified/path/to/manifest.json"` Per ulteriori informazioni su queste caratteristiche, consultare Funzionalità di recupero aggiuntive.	true
`AWS_APPCONFIG_EXTENSION_WAIT_ON_MANIFEST`	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true

Recupero di uno o più flag da una configurazione di feature flag

Per le configurazioni dei feature flag (configurazioni di tipo`AWS.AppConfig.FeatureFlags`), l'estensione Lambda consente di recuperare un singolo flag o un sottoinsieme di flag in una configurazione. Il recupero di uno o due flag è utile se la tua Lambda deve usare solo alcuni flag dal profilo di configurazione. I seguenti esempi utilizzano Python.

> **Note**
>
> La possibilità di chiamare un singolo flag di funzionalità o un sottoinsieme di flag in una configurazione è disponibile solo nella versione dell'estensione Agent AWS AppConfig Lambda 2.0.45 e successive.

È possibile recuperare i dati di AWS AppConfig configurazione da un endpoint HTTP locale. Per accedere a un flag specifico o a un elenco di flag, utilizzate il parametro di `?flag=flag_name` query per un AWS AppConfig profilo di configurazione.

Per accedere a un singolo flag e ai relativi attributi

```
import urllib.request

def lambda_handler(event, context):
    url = f'http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?flag=flag_name'
    config = urllib.request.urlopen(url).read()
    return config
```

Per accedere a più bandiere e ai relativi attributi

```
import urllib.request

def lambda_handler(event, context):
    url = f'http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?
flag=flag_name_one&flag=flag_name_two'
    config = urllib.request.urlopen(url).read()
    return config
```

Versioni disponibili dell'estensione AWS AppConfig Agent Lambda

Questo argomento include informazioni sulle versioni delle estensioni di AWS AppConfig Agent Lambda. L'estensione AWS AppConfig Agent Lambda supporta le funzioni Lambda sviluppate per le piattaforme x86-64 e ARM64 (Graviton2). Per funzionare correttamente, la funzione Lambda deve essere configurata per utilizzare lo specifico Amazon Resource Name (ARN) per il Regione AWS luogo in cui è attualmente ospitata. È possibile visualizzare Regione AWS i dettagli dell'ARN più avanti in questa sezione.

Important

Nota i seguenti dettagli importanti sull'estensione AWS AppConfig Agent Lambda.

- L'azione `GetConfiguration` API è stata dichiarata obsoleta il 28 gennaio 2022. Le chiamate per ricevere i dati di configurazione devono invece utilizzare le API `StartConfigurationSession` and`GetLatestConfiguration`. Se utilizzi una versione dell'estensione AWS AppConfig Agent Lambda creata prima del 28 gennaio 2022, potresti dover configurare l'autorizzazione per le nuove API. Per ulteriori informazioni, consulta Informazioni sul servizio AWS AppConfig Data Plane.
- AWS AppConfig supporta tutte le versioni elencate in. Versioni di estensione precedenti Ti consigliamo di eseguire periodicamente l'aggiornamento alla versione più recente per sfruttare i miglioramenti delle estensioni.

Argomenti

AWS AppConfig Note sulla versione di Agent Lambda Extension

La tabella seguente descrive le modifiche apportate alle versioni recenti dell'estensione AWS AppConfig Lambda.

Versione	Data di lancio	Note
2.0.358	12/01/2023	È stato aggiunto il supporto per le seguenti funzionalità di recupero: • Recupero di più account: utilizza AWS AppConfig Agent da un sistema

Versione	Data di lancio	Note
		primario o di recupero Account AWS per recuperare i dati di configurazione da più account fornitore. • Scrittura della copia della configurazione su disco: utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.
2.0.181	14/08/2023	Aggiunto il supporto per Israel (Tel Aviv) Regione AWS il-central-1.

Versione	Data di lancio	Note
2.0.165	21/02/2023	Correzioni di bug minori. Non si limita più l'uso dell'estensione a specifiche versioni di runtime tramite la console. AWS Lambda È stato aggiunto il supporto per quanto segue: Regioni AWS • Medio Oriente (EAU), me-central-1 • Asia Pacifico (Hyderabad), ap-south-2 • Asia Pacifico (Melbourne), ap-southeast-4 • Europa (Spagna), eu-south-2 • Europa (Zurigo), eu-central-2
2.0.122	23/08/2022	È stato aggiunto il supporto per un proxy di tunneling, che può essere configurato con le variabili di ambiente. `AWS_APPCONFIG_EXTENSION_PROXY_URL` `AWS_APPCONFIG_EXTENSION_PROXY_HEADERS` Aggiunto.NET 6 come runtime. Per ulteriori informazioni sulle variabili di ambiente, vedere[Configurazione dell'estensione AWS AppConfig Agent Lambda]().

Versione	Data di lancio	Note
2.0.58	05/03/2022	Supporto migliorato per i processori Graviton2 (ARM64) in Lambda.
2.0.45	15/03/2022	Aggiunto il supporto per la chiamata di un singolo flag di funzionalità. In precedenza, i clienti chiamavano i flag delle funzionalità raggruppati in un profilo di configurazione e dovevano analizzare la risposta lato client. Con questa versione, i clienti possono utilizzare un `flag=<flag-name>` parametro quando chiamano l'endpoint HTTP localhost per ottenere il valore di un singolo flag. È stato inoltre aggiunto il supporto iniziale per i processori Graviton2 (ARM64).

Individuazione del numero di versione dell'estensione Lambda

Utilizzare la procedura seguente per individuare il numero di versione dell'estensione AWS AppConfig Agent Lambda attualmente configurata. Per funzionare correttamente, la funzione Lambda deve essere configurata per utilizzare lo specifico Amazon Resource Name (ARN) per il Regione AWS luogo in cui è attualmente ospitata.

1. Accedi AWS Management Console e apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Scegli la funzione Lambda in cui desideri aggiungere il `AWS-AppConfig-Extension` layer.
3. Nella sezione Livelli, scegli Aggiungi un livello.
4. Nella sezione Scegli un livello, scegli AWS- AppConfig -Estensione dall'elenco dei AWS livelli.

5. Utilizzate l'elenco delle versioni per scegliere un numero di versione.
6. Scegli Aggiungi.
7. Utilizzate la scheda Test per testare la funzione.
8. Al termine del test, visualizza l'output del registro. Individua la versione dell'estensione AWS AppConfig Agent Lambda nella sezione Dettagli dell'esecuzione. Questa versione deve corrispondere agli URL richiesti per quella versione.

piattaforma x86-64

Quando aggiungi l'estensione come layer alla tua Lambda, devi specificare un ARN. Scegli un ARN dalla tabella seguente che corrisponda al Regione AWS luogo in cui hai creato la Lambda. Questi ARN sono per le funzioni Lambda sviluppate per la piattaforma x86-64.

Versione 2.0.358

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:128`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:93`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:141`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:161`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:93`

Regione	ARN
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:106`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:47`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:125`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:93`
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:98`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:159`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:83`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:44`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:76`

Regione	ARN
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:76`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:83`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:98`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:108`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:101`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:106`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:106`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:79`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:20`

Regione	ARN
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:107`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:47`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:128`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:83`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:22`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:49`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:85`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:54`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:54`

Piattaforma ARM64

Quando aggiungi l'estensione come layer alla tua Lambda, devi specificare un ARN. Scegli un ARN dalla tabella seguente che corrisponda al Regione AWS luogo in cui hai creato la Lambda. Questi ARN sono per le funzioni Lambda sviluppate per la piattaforma ARM64.

Versione 2.0.358

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:61`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:45`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:18`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:63`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:13`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:49`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension-Arm64:5`

Regione	ARN
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:63`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:45`
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:17`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:18`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:11`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension-Arm64:5`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:11`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:51`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:16`

Regione	ARN
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:16`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:58`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:49`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:16`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension-Arm64:5`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:49`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension-Arm64:5`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:16`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:11`

Regione	ARN
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension-Arm64:5`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:13`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension-Arm64:5`

Versioni di estensione precedenti

Questa sezione elenca gli ARN e Regioni AWS le versioni precedenti dell'estensione AWS AppConfig Lambda. Questo elenco non contiene informazioni per tutte le versioni precedenti dell'estensione AWS AppConfig Agent Lambda, ma verrà aggiornato quando verranno rilasciate nuove versioni.

Versioni di estensione precedenti (piattaforma x86-64)

Le tabelle seguenti elencano gli ARN e le Regioni AWS versioni precedenti dell'estensione AWS AppConfig Agent Lambda sviluppata per la piattaforma x86-64.

Data sostituita dalla nuova estensione: 12/01/2023

Versione 2.0.181

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:113`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:81`

Regione	ARN
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:124`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:146`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:81`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:93`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:32`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:110`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:81`
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:82`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:142`

Regione	ARN
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:73`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:29`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:68`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:68`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:73`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:84`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:93`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:86`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:91`

Regione	ARN
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:93`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:64`
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:5`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:94`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:32`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:113`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:73`
Israele (Tel Aviv)	`arn:aws:lambda:il-central-1:895787185223:layer:AWS-AppConfig-Extension:7`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:34`

Regione	ARN
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:73`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:46`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:46`

Data sostituita dalla nuova estensione: 14/08/2023

Versione 2.0.165

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:110`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:79`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:121`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:143`

Regione	ARN
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:79`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:91`
Europa (Zurigo)	`arn:aws:lambda:eu-central-2:758369105281:layer:AWS-AppConfig-Extension:29`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:108`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:79`
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:80`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:139`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:71`
Europa (Spagna)	`arn:aws:lambda:eu-south-2:586093569114:layer:AWS-AppConfig-Extension:26`

Regione	ARN
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:66`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:66`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:71`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:82`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:91`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:84`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:89`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:91`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:60`

Regione	ARN
Asia Pacifico (Melbourne)	`arn:aws:lambda:ap-southeast-4:307021474294:layer:AWS-AppConfig-Extension:2`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:92`
Asia Pacific (Hyderabad)	`arn:aws:lambda:ap-south-2:489524808438:layer:AWS-AppConfig-Extension:29`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:110`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:71`
Medio Oriente (Emirati Arabi Uniti)	`arn:aws:lambda:me-central-1:662846165436:layer:AWS-AppConfig-Extension:31`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:71`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:44`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:44`

Data sostituita dalla nuova estensione: 21/02/2023

Versione 2.0.122

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:82`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:59`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:93`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:114`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:59`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:70`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:82`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:59`

Regione	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:60`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:111`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:54`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:52`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:52`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:54`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:62`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:70`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:59`

Regione	ARN
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:64`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:70`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:37`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:71`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:82`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:54`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:54`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:29`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:29`

Data sostituita dalla nuova estensione: 23/08/2022

Versione 2.0.58

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:69`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:50`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:78`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:101`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:50`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:59`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:69`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:50`

Regione	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:51`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:98`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:47`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:46`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:46`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:47`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:49`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:59`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:46`

Regione	ARN
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:51`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:59`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:24`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:60`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:69`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:47`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:47`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:23`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:23`

Data sostituita dalla nuova estensione: 21/04/2022

Versione 2.0.45

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:68`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:49`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:77`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:100`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:49`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:58`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:68`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:49`

Regione	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:50`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:97`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:46`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:45`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:45`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:46`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:48`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:58`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:45`

Regione	ARN
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:50`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:58`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:23`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:59`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:68`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:46`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:46`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:22`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:22`

Data sostituita dalla nuova estensione: 15/03/2022

Versione 2.0.30

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension:61`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension:47`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension:61`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension:89`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension:47`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension:54`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension:59`
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension:47`

Regione	ARN
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension:48`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension:86`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension:44`
Cina (Pechino)	`arn:aws-cn:lambda:cn-north-1:615057806174:layer:AWS-AppConfig-Extension:43`
Cina (Ningxia)	`arn:aws-cn:lambda:cn-northwest-1:615084187847:layer:AWS-AppConfig-Extension:43`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension:44`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension:45`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension:42`
Asia Pacific (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension:54`

Regione	ARN
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension:45`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension:54`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension:13`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension:55`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension:61`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension:44`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension:44`
AWS GovCloud (Stati Uniti orientali)	`arn:aws-us-gov:lambda:us-gov-east-1:946561847325:layer:AWS-AppConfig-Extension:20`
AWS GovCloud (Stati Uniti occidentali)	`arn:aws-us-gov:lambda:us-gov-west-1:946746059096:layer:AWS-AppConfig-Extension:20`

Versioni di estensione precedenti (piattaforma ARM64)

Le tabelle seguenti elencano gli ARN e le Regioni AWS versioni precedenti dell'estensione AWS AppConfig Agent Lambda sviluppata per la piattaforma ARM64.

Data sostituita dalla nuova estensione: 12/01/2023

Versione 2.0.181

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:46`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:33`
Stati Uniti occidentali (California settentrionale)	`arn:aws:lambda:us-west-1:958113053741:layer:AWS-AppConfig-Extension-Arm64:1`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:48`
Canada (Centrale)	`arn:aws:lambda:ca-central-1:039592058896:layer:AWS-AppConfig-Extension-Arm64:1`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:36`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:48`

Regione	ARN
Europe (London)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:33`
Europe (Paris)	`arn:aws:lambda:eu-west-3:493207061005:layer:AWS-AppConfig-Extension-Arm64:1`
Europe (Stockholm)	`arn:aws:lambda:eu-north-1:646970417810:layer:AWS-AppConfig-Extension-Arm64:1`
Europa (Milano)	`arn:aws:lambda:eu-south-1:203683718741:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Hong Kong)	`arn:aws:lambda:ap-east-1:630222743974:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:37`
Asia Pacifico (Seul)	`arn:aws:lambda:ap-northeast-2:826293736237:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Osaka-Locale)	`arn:aws:lambda:ap-northeast-3:706869817123:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:43`

Regione	ARN
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:36`
Asia Pacifico (Giacarta)	`arn:aws:lambda:ap-southeast-3:418787028745:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:36`
Sud America (San Paolo)	`arn:aws:lambda:sa-east-1:000010852771:layer:AWS-AppConfig-Extension-Arm64:1`
Africa (Città del Capo)	`arn:aws:lambda:af-south-1:574348263942:layer:AWS-AppConfig-Extension-Arm64:1`
Medio Oriente (Bahrein)	`arn:aws:lambda:me-south-1:559955524753:layer:AWS-AppConfig-Extension-Arm64:1`

Data sostituita dalla nuova estensione: 30/03/2023

Versione 2.0.165

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:43`

Regione	ARN
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:31`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:45`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:34`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:46`
Europa (Londra)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:31`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:35`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:41`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:34`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:34`

Data sostituita dalla nuova estensione: 21/02/2023

Versione 2.0.122

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:15`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:11`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:16`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:13`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:20`
Europa (Londra)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:11`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:15`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:16`

Regione	ARN
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:13`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:13`

Data sostituita dalla nuova estensione: 23/08/2022

Versione 2.0.58

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:2`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:2`
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:3`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:2`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:7`

Regione	ARN
Europa (Londra)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:2`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:2`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:3`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:2`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:2`

Data sostituita dalla nuova estensione: 21/04/2022

Versione 2.0.45

Regione	ARN
Stati Uniti orientali (Virginia settentrionale)	`arn:aws:lambda:us-east-1:027255383542:layer:AWS-AppConfig-Extension-Arm64:1`
Stati Uniti orientali (Ohio)	`arn:aws:lambda:us-east-2:728743619870:layer:AWS-AppConfig-Extension-Arm64:1`

Regione	ARN
US West (Oregon)	`arn:aws:lambda:us-west-2:359756378197:layer:AWS-AppConfig-Extension-Arm64:2`
Europa (Francoforte)	`arn:aws:lambda:eu-central-1:066940009817:layer:AWS-AppConfig-Extension-Arm64:1`
Europa (Irlanda)	`arn:aws:lambda:eu-west-1:434848589818:layer:AWS-AppConfig-Extension-Arm64:6`
Europa (Londra)	`arn:aws:lambda:eu-west-2:282860088358:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Tokyo)	`arn:aws:lambda:ap-northeast-1:980059726660:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Singapore)	`arn:aws:lambda:ap-southeast-1:421114256042:layer:AWS-AppConfig-Extension-Arm64:2`
Asia Pacifico (Sydney)	`arn:aws:lambda:ap-southeast-2:080788657173:layer:AWS-AppConfig-Extension-Arm64:1`
Asia Pacifico (Mumbai)	`arn:aws:lambda:ap-south-1:554480029851:layer:AWS-AppConfig-Extension-Arm64:1`

Utilizzo di un'immagine del contenitore per aggiungere l'estensione AWS AppConfig Agent Lambda

Puoi impacchettare l'estensione AWS AppConfig Agent Lambda come immagine del contenitore per caricarla nel registro dei container ospitato su Amazon Elastic Container Registry (Amazon ECR).

Per aggiungere l'estensione AWS AppConfig Agent Lambda come immagine del contenitore Lambda

1. Inserisci Regione AWS e l'Amazon Resource Name (ARN) nel campo AWS Command Line Interface (AWS CLI) come mostrato di seguito. Sostituisci il valore Region e ARN con la tua regione e l'ARN corrispondente per recuperare una copia del layer Lambda.

```
aws lambda get-layer-version-by-arn \
  --region us-east-1 \
  --arn arn:aws:lambda::layer:AWS-AppConfig-Extension:00
```

   Di seguito è riportata la risposta.

```
{
  "LayerVersionArn": "arn:aws:lambda::layer:AWS-AppConfig-Extension:00",
  "Description": "AWS AppConfig extension: Use dynamic configurations deployed via
 AWS AppConfig for your AWS Lambda functions",
  "CreatedDate": "2021-04-01T02:37:55.339+0000",
  "LayerArn": "arn:aws:lambda::layer:AWS-AppConfig-Extension",

  "Content": {
    "CodeSize": 5228073,
    "CodeSha256": "8otOgbLQbexpUm3rKlMhvcE6Q5TvwcLCKrc4Oe+vmMY=",
    "Location" : "S3-Bucket-Location-URL"
  },

  "Version": 30,
  "CompatibleRuntimes": [
    "python3.8",
    "python3.7",
    "nodejs12.x",
    "ruby2.7"
  ],
}
```

2. Nella risposta precedente, il valore restituito `Location` è l'URL del bucket Amazon Simple Storage Service (Amazon S3) che contiene l'estensione Lambda. Incolla l'URL nel tuo browser web per scaricare il file.zip con estensione Lambda.

   > **Note**
   >
   > L'URL del bucket Amazon S3 è disponibile solo per 10 minuti.

   (Facoltativo) In alternativa, puoi anche usare il seguente `curl` comando per scaricare l'estensione Lambda.

   ```
   curl -o extension.zip "S3-Bucket-Location-URL"
   ```

   (Facoltativo) In alternativa, è possibile combinare i passaggi 1 e 2 per recuperare l'ARN e scaricare il file con estensione.zip tutto in una volta.

   ```
   aws lambda get-layer-version-by-arn \
     --arn arn:aws:lambda::layer:AWS-AppConfig-Extension:00 \
     | jq -r '.Content.Location' \
     | xargs curl -o extension.zip
   ```

3. Aggiungi le seguenti righe `Dockerfile` per aggiungere l'estensione all'immagine del contenitore.

   ```
   COPY extension.zip extension.zip
   RUN yum install -y unzip \
     && unzip extension.zip /opt \
     && rm -f extension.zip
   ```

4. Assicurati che il ruolo di esecuzione della funzione Lambda abbia il set di autorizzazioni appconfig:. GetConfiguration

Esempio

Questa sezione include un esempio per abilitare l'estensione AWS AppConfig Agent Lambda su una funzione Python Lambda basata su immagini del contenitore.

1. Creane una `Dockerfile` che sia simile alla seguente.

```
FROM public.ecr.aws/lambda/python:3.8 AS builder
COPY extension.zip extension.zip
RUN yum install -y unzip \
  && unzip extension.zip -d /opt \
  && rm -f extension.zip

FROM public.ecr.aws/lambda/python:3.8
COPY --from=builder /opt /opt
COPY index.py ${LAMBDA_TASK_ROOT}
CMD [ "index.handler" ]
```

2. Scaricate il livello di estensione nella stessa directory di`Dockerfile`.

```
aws lambda get-layer-version-by-arn \
  --arn arn:aws:lambda::layer:AWS-AppConfig-Extension:00 \
  | jq -r '.Content.Location' \
  | xargs curl -o extension.zip
```

3. Crea un file Python denominato `index.py` nella stessa directory di. `Dockerfile`

```
import urllib.request

def handler(event, context):
    return {
       # replace parameters here with your application, environment, and
 configuration names
       'configuration': get_configuration('myApp', 'myEnv', 'myConfig'),
    }

def get_configuration(app, env, config):
    url = f'http://localhost:2772/applications/{app}/environments/{env}/
configurations/{config}'
    return urllib.request.urlopen(url).read()
```

4. Esegui i passaggi seguenti per creare l'`docker`immagine e caricarla su Amazon ECR.

```
// set environment variables
export ACCOUNT_ID = <YOUR_ACCOUNT_ID>
export REGION = <AWS_REGION>

// create an ECR repository
aws ecr create-repository --repository-name test-repository
```

```
// build the docker image
docker build -t test-image .

// sign in to AWS
aws ecr get-login-password \
  | docker login \
  --username AWS \
  --password-stdin "$ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com"

// tag the image
docker tag test-image:latest "$ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com/test-
repository:latest"

// push the image to ECR
docker push "$ACCOUNT_ID.dkr.ecr.$REGION.amazonaws.com/test-repository:latest"
```

5. Usa l'immagine Amazon ECR che hai creato sopra per creare la funzione Lambda. Per ulteriori informazioni su una funzione Lambda come contenitore, consulta Creare una funzione Lambda definita come immagine contenitore.
6. Assicurati che il ruolo di esecuzione della funzione Lambda abbia il set di autorizzazioni appconfig:. GetConfiguration

Integrazione con OpenAPI

È possibile utilizzare la seguente specifica YAML per OpenAPI per creare un SDK utilizzando uno strumento come OpenAPI Generator. È possibile aggiornare questa specifica per includere valori codificati per l'applicazione, l'ambiente o la configurazione. Puoi anche aggiungere percorsi aggiuntivi (se disponi di più tipi di configurazione) e includere schemi di configurazione per generare modelli tipizzati specifici della configurazione per i tuoi client SDK. Per ulteriori informazioni su OpenAPI (noto anche come Swagger), consulta la specifica OpenAPI.

```
openapi: 3.0.0
info:
  version: 1.0.0
  title: AppConfig Agent Lambda extension API
  description: An API model for the AppConfig Agent Lambda extension.
servers:
  - url: https://localhost:{port}/
    variables:
```

```
      port:
        default:
          '2772'
paths:
  /applications/{Application}/environments/{Environment}/configurations/
{Configuration}:
    get:
      operationId: getConfiguration
      tags:
        - configuration
      parameters:
        - in: path
          name: Application
          description: The application for the configuration to get. Specify either the
 application name or the application ID.
          required: true
          schema:
            type: string
        - in: path
          name: Environment
          description: The environment for the configuration to get. Specify either the
 environment name or the environment ID.
          required: true
          schema:
            type: string
        - in: path
          name: Configuration
          description: The configuration to get. Specify either the configuration name
 or the configuration ID.
          required: true
          schema:
            type: string
      responses:
        200:
          headers:
            ConfigurationVersion:
              schema:
                type: string
          content:
            application/octet-stream:
              schema:
                type: string
                format: binary
          description: successful config retrieval
```

```
        400:
          description: BadRequestException
          content:
            application/text:
              schema:
                $ref: '#/components/schemas/Error'
        404:
          description: ResourceNotFoundException
          content:
            application/text:
              schema:
                $ref: '#/components/schemas/Error'
        500:
          description: InternalServerException
          content:
            application/text:
              schema:
                $ref: '#/components/schemas/Error'
        502:
          description: BadGatewayException
          content:
            application/text:
              schema:
                $ref: '#/components/schemas/Error'
        504:
          description: GatewayTimeoutException
          content:
            application/text:
              schema:
                $ref: '#/components/schemas/Error'

components:
  schemas:
    Error:
      type: string
      description: The response error
```

Recupero dei dati di configurazione dalle istanze Amazon EC2

Puoi integrarti AWS AppConfig con le applicazioni in esecuzione sulle tue istanze Amazon Elastic Compute Cloud (Amazon EC2) Linux utilizzando Agent. AWS AppConfig L'agente migliora l'elaborazione e la gestione delle applicazioni nei seguenti modi:

- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale dei dati di configurazione. Estraendo i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati. *
- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei AWS AppConfig flag di funzionalità.
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.

*AWS AppConfig L'agente memorizza i dati nella cache la prima volta che il servizio recupera i dati di configurazione. Per questo motivo, la prima chiamata per recuperare i dati è più lenta delle chiamate successive.

Argomenti

Passaggio 1: (Obbligatorio) Creazione di risorse e configurazione delle autorizzazioni

Per l'integrazione AWS AppConfig con le applicazioni in esecuzione sulle tue istanze Amazon EC2, devi creare AWS AppConfig artefatti e dati di configurazione, inclusi flag di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig.

Per recuperare i dati di configurazione ospitati da AWS AppConfig, le applicazioni devono essere configurate con accesso al piano dati. AWS AppConfig Per consentire l'accesso alle applicazioni, aggiorna la policy di autorizzazione IAM assegnata al ruolo dell'istanza Amazon EC2. In particolare, è necessario aggiungere le `appconfig:GetLatestConfiguration` azioni `appconfig:StartConfigurationSession` e alla policy. Ecco un esempio:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "appconfig:StartConfigurationSession",
                "appconfig:GetLatestConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Per ulteriori informazioni sull'aggiunta di autorizzazioni a una policy, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella IAM User Guide.

Fase 2: (Obbligatorio) Installazione e avvio AWS AppConfig dell'agente sulle istanze Amazon EC2

AWS AppConfig L'agente è ospitato in un bucket Amazon Simple Storage Service (Amazon S3) gestito da. AWS Usa la seguente procedura per installare la versione più recente dell'agente sulla tua istanza Linux. Se l'applicazione è distribuita su più istanze, è necessario eseguire questa procedura su ogni istanza che ospita l'applicazione.

> **Note**
>
> Prendi nota delle seguenti informazioni:
>
> - AWS AppConfig L'agente è disponibile per i sistemi operativi Linux che eseguono la versione del kernel 4.15 o successiva. I sistemi basati su Debian, come Ubuntu, non sono supportati.
> - L'agente supporta le architetture x86_64 e ARM64.

- Per le applicazioni distribuite, consigliamo di aggiungere i comandi di installazione e avvio ai dati utente Amazon EC2 del gruppo Auto Scaling. In tal caso, ogni istanza esegue i comandi automaticamente. Per maggiori informazioni, consulta Esecuzione di comandi sull'istanza Linux all'avvio nella Guida dell'utente di Amazon EC2 per le istanze Linux. Inoltre, consulta Tutorial: Configura i dati utente per recuperare lo stato del ciclo di vita di destinazione tramite i metadati dell'istanza nella Amazon EC2 Auto Scaling User Guide.
- Le procedure illustrate in questo argomento descrivono come eseguire azioni come l'installazione dell'agente accedendo all'istanza per eseguire il comando. È possibile eseguire i comandi da un computer client locale e indirizzare una o più istanze utilizzando Run Command, che è una funzionalità di. AWS Systems Manager Per ulteriori informazioni, consulta Run Command AWS Systems Manager nella Guida per l'utente AWS Systems Manager .
- AWS AppConfig L'agente sulle istanze Linux di Amazon EC2 è un servizio. `systemd`

Per installare e avviare AWS AppConfig Agent su un'istanza

1. Accedi alla tua istanza Linux.
2. Apri un terminale ed esegui il seguente comando con i permessi di amministratore per le architetture x86_64:

   ```
   sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/x86_64/latest/aws-appconfig-agent.rpm
   ```

   Per le architetture ARM64, esegui il seguente comando:

   ```
   sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/arm64/latest/aws-appconfig-agent.rpm
   ```

   Se desideri installare una versione specifica di AWS AppConfig Agent, sostituisci `latest` l'URL con un numero di versione specifico. Ecco un esempio per x86_64:

   ```
   sudo yum install https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/linux/x86_64/2.0.2/aws-appconfig-agent.rpm
   ```

3. Esegui il comando seguente per avviare l'agente:

```
sudo systemctl start aws-appconfig-agent
```

4. Esegui il comando seguente per verificare che l'agente sia in esecuzione:

```
sudo systemctl status aws-appconfig-agent
```

In caso di successo, il comando restituisce informazioni come le seguenti:

```
aws-appconfig-agent.service - aws-appconfig-agent
  ...
  Active: active (running) since Mon 2023-07-26 00:00:00 UTC; 0s ago
  ...
```

> **Note**
>
> Per arrestare l'agente, esegui questo comando:
>
> ```
> sudo systemctl stop aws-appconfig-agent
> ```

Passaggio 3: (Facoltativo, ma consigliato) Invio dei file di registro ai CloudWatch registri

Per impostazione predefinita, AWS AppConfig Agent pubblica i log su STDERR. Systemd reindirizza STDOUT e STDERR per tutti i servizi in esecuzione sull'istanza Linux al journal systemd. È possibile visualizzare e gestire i dati di registro nel journal systemd se si esegue Agent solo su una o due istanze. AWS AppConfig Una soluzione migliore, una soluzione che consigliamo vivamente per le applicazioni distribuite, è scrivere file di registro su disco e quindi utilizzare Amazon CloudWatch Agent per caricare i dati di registro sul AWS cloud. Inoltre, puoi configurare l' CloudWatch agente per eliminare i vecchi file di registro dall'istanza, in modo da evitare che l'istanza esaurisca lo spazio su disco.

Per abilitare la registrazione su disco, è necessario impostare la variabile di `LOG_PATH` ambiente, come descritto inFase 4: (Facoltativo) Utilizzo delle variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2.

Per iniziare a usare l' CloudWatch agente, consulta Raccogli metriche e log dalle istanze Amazon EC2 e dai server locali con l' CloudWatch agente nella Amazon User Guide. CloudWatch È possibile utilizzare Quick Setup, una funzionalità di Systems Manager per installare rapidamente l' CloudWatch agente. Per ulteriori informazioni, vedere Quick Setup Host Management nella Guida AWS Systems Manager per l'utente.

> **Warning**
>
> Se si sceglie di scrivere i file di registro su disco senza utilizzare l' CloudWatch agente, è necessario eliminare i vecchi file di registro. AWS AppConfig L'agente ruota automaticamente i file di registro ogni ora. Se non elimini i vecchi file di registro, l'istanza può esaurire lo spazio su disco.

Dopo aver installato l' CloudWatch agente sull'istanza, crea un file di configurazione CloudWatch dell'agente. Il file di configurazione spiega all' CloudWatch agente come lavorare con i file di registro AWS AppConfig dell'agente. Per ulteriori informazioni sulla creazione di un file di configurazione CloudWatch dell'agente, vedere Creare il file di configurazione dell' CloudWatch agente.

Aggiungi la `logs` sezione seguente al file di configurazione dell' CloudWatch agente sull'istanza e salva le modifiche:

```
"logs": {
  "logs_collected": {
    "files": {
      "collect_list": [
        {
          "file_path": "/path_you_specified_for_logging",
          "log_group_name": "${YOUR_LOG_GROUP_NAME}/aws-appconfig-agent.log",
          "auto_removal": true
        },
        ...
      ]
    },
    ...
  },
  ...
}
```

Se il valore di `auto_removal` è`true`, l' CloudWatch agente elimina automaticamente i file di registro AWS AppConfig dell'agente ruotati.

Fase 4: (Facoltativo) Utilizzo delle variabili di ambiente per configurare AWS AppConfig Agent for Amazon EC2

Puoi configurare AWS AppConfig Agent for Amazon EC2 utilizzando variabili di ambiente. Per impostare le variabili di ambiente per un `systemd` servizio, crei un file di unità drop-in. L'esempio seguente mostra come creare un file di unità drop-in su cui impostare il livello di registrazione dell' AWS AppConfig agente. `DEBUG`

Esempio di come creare un file di unità drop-in per le variabili di ambiente

1. Accedi alla tua istanza Linux.
2. Apri un terminale ed esegui il seguente comando con i permessi di amministratore. Il comando crea una directory di configurazione:

   ```
   sudo mkdir /etc/systemd/system/aws-appconfig-agent.service.d
   ```

3. Esegui il comando seguente per creare il file dell'unità drop-in. Sostituisci *`file_name con un nome`* per il file. L'estensione deve essere: `.conf`

   ```
   sudo touch /etc/systemd/system/aws-appconfig-agent.service.d/file_name.conf
   ```

4. Inserisci le informazioni nel file dell'unità drop-in. L'esempio seguente aggiunge una `Service` sezione che definisce una variabile di ambiente. L'esempio imposta il livello di registro AWS AppConfig dell'agente su`DEBUG`.

   ```
   [Service]
   Environment=LOG_LEVEL=DEBUG
   ```

5. Eseguite il comando seguente per ricaricare la configurazione systemd:

   ```
   sudo systemctl daemon-reload
   ```

6. Eseguite il seguente comando per riavviare AWS AppConfig l'agente:

   ```
   sudo systemctl restart aws-appconfig-agent
   ```

Puoi configurare AWS AppConfig Agent for Amazon EC2 specificando le seguenti variabili di ambiente in un file di unità drop-in.

Variabile di ambiente	Informazioni	Valore predefinito

`ACCESS_TOKEN`	Questa variabile di ambiente definisce un token che deve essere fornito quando si richiedono i dati di configura zione dal server HTTP dell'agente. Il valore del token deve essere impostato nell'inte stazione di autorizzazione della richiesta HTTP con un tipo di autorizzazione di. `Bearer` Ecco un esempio.
	`GET /applications/my_a pp/... Host: localhost:2772`
	`Authorization: Bearer <token value>`
		Nessuno

`BACKUP_DIRECTORY`	Questa variabile di ambiente consente all' AWS AppConfig agente di salvare un backup di ogni configurazione recuperat a nella directory specificata. **Important** Le configurazioni di cui è stato eseguito il backup su disco non sono crittogra fate. Se la configura	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
	zione contiene dati sensibili, si AWS AppConfig consiglia di applicare il principio del privilegio minimo con le autorizzazioni del file system. Per ulteriori informazioni, consulta Sicurezza in AWS AppConfig.	
`HTTP_PORT`	Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP per l'agente.	2772
`LOG_LEVEL`	Questa variabile di ambiente specifica il livello di dettaglio registrato dall'agente. Ogni livello include il livello corrente e tutti i livelli superiori. Le variabili distinguono tra maiuscole e minuscole. Dal più dettaglia to al meno dettagliato, i livelli di registro sono: `debug`,`info`,`warn`,`error`, e`none`. `Debug`include informazioni dettagliate, incluse informazioni sulla tempistica, sull'agente.	`info`

Variabile di ambiente	Informazioni	Valore predefinito
`LOG_PATH`	La posizione su disco in cui vengono scritti i log. Se non specificato, i log vengono scritti su stderr.	Nessuno
`MANIFEST`	Questa variabile di ambiente configura AWS AppConfig Agent per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. È possibile inserire uno dei seguenti valori: • `"app:env:manifest-config"` • `"file:/fully/qualified/path/to/manifest.json"` Per ulteriori informazioni su queste caratteristiche, consultare Funzionalità di recupero aggiuntive.	true
`MAX_CONNECTIONS`	Questa variabile di ambiente configura il numero massimo di connessioni utilizzate dall'agente per recuperare le configurazioni. AWS AppConfig	3

Variabile di ambiente	Informazioni	Valore predefinito
`POLL_INTERVAL`	Questa variabile di ambiente controlla la frequenza con cui l'agente richiede dati di configurazione aggiornati. AWS AppConfig È possibile specificare un numero di secondi per l'intervallo. È inoltre possibile specificare un numero con un'unità di tempo: s per secondi, m per minuti e h per ore. Se non viene specificata un'unità, l'agente utilizza come impostazi one predefinita i secondi. Ad esempio, 60, 60 e 1 m generano lo stesso intervallo di sondaggio.	45 secondi
`PREFETCH_LIST`	Questa variabile di ambiente specifica i dati di configura zione richiesti dall'agente non AWS AppConfig appena viene avviato.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
PRELOAD_BACKUPS	Se impostato su`true`, AWS AppConfig l'agente carica i backup di configurazione presenti BACKUP_DIRECTORY nella memoria e verifica immediatamente se esiste una versione più recente del servizio. Se impostato su`false`, l' AWS AppConfig agente carica i contenuti da un backup di configurazione solo se non è in grado di recuperare i dati di configurazione dal servizio, ad esempio se c'è un problema con la rete.	true
PROXY_HEADERS	Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente. PROXY_URL Il valore è un elenco di intestazioni separate da virgole. Ogni intestazione utilizza il seguente modulo. `"header: value"`	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
PROXY_URL	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall'agente a Servizi AWS, incluso. AWS AppConfig HTTPSe gli HTTP URL sono supportati.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
`REQUEST_TIMEOUT`	Questa variabile di ambiente controlla la quantità di tempo da cui l'agente attende una risposta. AWS AppConfig Se il servizio non risponde, la richiesta ha esito negativo. Se la richiesta riguarda il recupero iniziale dei dati, l'agente restituisce un errore all'applicazione. Se il timeout si verifica durante un controllo in background per verificare la presenza di dati aggiornati, l'agente registra l'errore e riprova dopo un breve ritardo. È possibile specificare il numero di millisecondi per il timeout. È inoltre possibile specificare un numero con un'unità di tempo: ms per millisecondi e s per secondi. Se non viene specificata un'unità, l'agente utilizza come impostazione predefinita i millisecondi. Ad esempio, 5000, 5000 ms e 5 secondi generano lo stesso valore di timeout della richiesta.	3000 millisecondi

Variabile di ambiente	Informazioni	Valore predefinito
ROLE_ARN	Questa variabile di ambiente specifica l'Amazon Resource Name (ARN) di un ruolo IAM. AWS AppConfig L'agente assume questo ruolo per recuperare i dati di configura zione.	Nessuno
ROLE_EXTERNAL_ID	Questa variabile di ambiente specifica l'ID esterno da utilizzare con il ruolo ARN assunto.	Nessuno
ROLE_SESSION_NAME	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno
SERVICE_REGION	Questa variabile di ambiente specifica un'alternativa Regione AWS utilizzata da AWS AppConfig Agent per chiamare il servizio. AWS AppConfig Se non viene definita, l'agente tenta di determinare la regione corrente. In caso contrario, l'agente non si avvia.	Nessuno
WAIT_ON_MANIFEST	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true

Fase 5: (Obbligatorio) Recupero dei dati di configurazione

È possibile recuperare i dati di configurazione dall' AWS AppConfig agente utilizzando una chiamata HTTP localhost. Gli esempi seguenti vengono utilizzati `curl` con un client HTTP. È possibile chiamare l'agente utilizzando qualsiasi client HTTP disponibile supportato dal linguaggio dell'applicazione o dalle librerie disponibili, incluso un AWS SDK.

Per recuperare il contenuto completo di qualsiasi configurazione distribuita

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name"
```

Per recuperare un singolo flag e i relativi attributi da una AWS AppConfig configurazione di tipo **`Feature Flag`**

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?flag=flag_name"
```

Per accedere a più flag e ai relativi attributi da una AWS AppConfig configurazione di tipo **`Feature Flag`**

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?
flag=flag_name_one&flag=flag_name_two"
```

Passaggio 6 (facoltativo, ma consigliato): Automatizzazione degli aggiornamenti all'agente AWS AppConfig

AWS AppConfig L'agente viene aggiornato periodicamente. Per assicurarti di eseguire la versione più recente di AWS AppConfig Agent sulle tue istanze, ti consigliamo di aggiungere i seguenti comandi ai dati utente di Amazon EC2. Puoi aggiungere i comandi ai dati utente sull'istanza o sul gruppo Auto Scaling EC2. Lo script installa e avvia la versione più recente dell'agente ogni volta che un'istanza viene avviata o riavviata.

```
#!/bin/bash
# install the latest version of the agent
yum install -y https://s3.amazonaws.com/aws-appconfig-downloads/aws-appconfig-agent/
linux/x86_64/latest/aws-appconfig-agent.rpm
# optional: configure the agent
```

```
mkdir /etc/systemd/system/aws-appconfig-agent.service.d
echo "${MY_AGENT_CONFIG}" > /etc/systemd/system/aws-appconfig-agent.service.d/
overrides.conf
systemctl daemon-reload
# start the agent
systemctl start aws-appconfig-agent
```

Recupero dei dati di configurazione da Amazon ECS e Amazon EKS

Puoi effettuare l'integrazione AWS AppConfig con Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS) utilizzando Agent. AWS AppConfig L'agente funziona come un contenitore secondario che funziona insieme alle applicazioni container Amazon ECS e Amazon EKS. L'agente migliora l'elaborazione e la gestione delle applicazioni containerizzate nei seguenti modi:

- L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale di dati di configurazione. Estraendo i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati. *
- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei AWS AppConfig flag di funzionalità.
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione containerizzata può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.
- AWS AppConfig L'agente aggiorna i dati di configurazione nei contenitori senza dover riavviare o riciclare tali contenitori.

*AWS AppConfig L'agente memorizza i dati nella cache la prima volta che il servizio recupera i dati di configurazione. Per questo motivo, la prima chiamata per recuperare i dati è più lenta delle chiamate successive.

Argomenti

Prima di iniziare

Per l'integrazione AWS AppConfig con le applicazioni container, è necessario creare AWS AppConfig artefatti e dati di configurazione, inclusi flag di funzionalità o dati di configurazione in formato libero. Per ulteriori informazioni, consulta Creazione di flag di funzionalità e dati di configurazione in formato libero in AWS AppConfig.

Per recuperare i dati di configurazione ospitati da AWS AppConfig, le applicazioni container devono essere configurate con accesso al piano dati. AWS AppConfig Per consentire l'accesso alle applicazioni, aggiorna la policy di autorizzazione IAM utilizzata dal ruolo IAM del servizio container. In particolare, devi aggiungere le `appconfig:GetLatestConfiguration` azioni `appconfig:StartConfigurationSession` e alla policy. I ruoli IAM del servizio container includono i seguenti:

- Il ruolo dell'attività di Amazon ECS
- Il ruolo del nodo Amazon EKS
- Il ruolo di esecuzione del AWS Fargate (Fargate) pod (se i contenitori Amazon EKS utilizzano Fargate per l'elaborazione del calcolo)

Per ulteriori informazioni sull'aggiunta di autorizzazioni a una policy, consulta Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'utente IAM.

Avvio dell' AWS AppConfig agente per l'integrazione con Amazon ECS

Il contenitore sidecar AWS AppConfig Agent è automaticamente disponibile nel tuo ambiente Amazon ECS. Per utilizzare il contenitore AWS AppConfig Agent sidecar, devi avviarlo.

Per avviare Amazon ECS (console)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.

2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli la definizione dell'attività per la tua applicazione, quindi seleziona la revisione più recente.
4. Scegli Crea nuova revisione, Crea nuova revisione.
5. Scegli Aggiungi altri contenitori.
6. In Nome, inserisci un nome univoco per il contenitore AWS AppConfig dell'agente.
7. Per l'URI dell'immagine, inserisci: **`public.ecr.aws/aws-appconfig/aws-appconfig-agent:2.x`**
8. Per Essential container, scegli Sì.
9. Nella sezione Mappature delle porte, scegli Aggiungi mappatura delle porte.
10. Per Container port, inserisci. **`2772`**

> **Note**
>
> AWS AppConfig Per impostazione predefinita, l'agente viene eseguito sulla porta 2772. È possibile specificare una porta diversa.

11. Scegli Crea. Amazon ECS crea una nuova revisione del contenitore e ne visualizza i dettagli.
12. Nel pannello di navigazione, scegli Clusters, quindi scegli il tuo cluster di applicazioni nell'elenco.
13. Nella scheda Servizi, seleziona il servizio per la tua applicazione.
14. Scegli Aggiorna.
15. In Configurazione di distribuzione, per Revisione, scegli la revisione più recente.
16. Scegli Aggiorna. Amazon ECS implementa la definizione di attività più recente.
17. Al termine della distribuzione, puoi verificare che l' AWS AppConfig agente sia in esecuzione nella scheda Configurazione e attività. Nella scheda Attività, scegli l'attività in esecuzione.
18. Nella sezione Contenitori, verifica che il contenitore AWS AppConfig dell'agente sia elencato.
19. Per verificare che AWS AppConfig l'agente sia stato avviato, scegli la scheda Registri. Individua un'istruzione come la seguente per il contenitore dell' AWS AppConfig agente: `[appconfig agent] 1970/01/01 00:00:00 INFO serving on localhost:2772`

> **Note**
>
> È possibile regolare il comportamento predefinito di AWS AppConfig Agent inserendo o modificando le variabili di ambiente. Per informazioni sulle variabili di ambiente disponibili,

vedereUtilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS. Per informazioni su come modificare le variabili di ambiente in Amazon ECS, consulta Passing environment variables to a container nella Amazon Elastic Container Service Developer Guide.

Avvio dell' AWS AppConfig agente per l'integrazione con Amazon EKS

Il contenitore sidecar AWS AppConfig Agent è automaticamente disponibile nel tuo ambiente Amazon EKS. Per utilizzare il contenitore AWS AppConfig Agent sidecar, devi avviarlo. La procedura seguente descrive come utilizzare lo strumento a riga di `kubectl` comando di Amazon EKS per apportare modifiche al `kubeconfig` file per la tua applicazione contenitore. Per ulteriori informazioni sulla creazione o la modifica di un `kubeconfig` file, consulta Creazione o aggiornamento di un file kubeconfig per un cluster Amazon EKS nella Guida per l'utente di Amazon EKS.

Per avviare AWS AppConfig Agent (strumento da riga di comando kubectl)

1. Apri il `kubeconfig` file e verifica che l'applicazione Amazon EKS sia in esecuzione come distribuzione a contenitore singolo. Il contenuto del file dovrebbe essere simile al seguente.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: my-namespace
  labels:
    app: my-application-label
spec:
  replicas: 1
  selector:
    matchLabels:
      app: my-application-label
  template:
    metadata:
      labels:
        app: my-application-label
    spec:
      containers:
      - name: my-app
        image: my-repo/my-image
        imagePullPolicy: IfNotPresent
```

2. Aggiungi i dettagli della definizione del contenitore AWS AppConfig Agent al tuo file di distribuzione YAML.

```
- name: appconfig-agent
        image: public.ecr.aws/aws-appconfig/aws-appconfig-agent:2.x
        ports:
        - name: http
          containerPort: 2772
          protocol: TCP
        env:
        - name: SERVICE_REGION
          value: region
        imagePullPolicy: IfNotPresent
```

> **Note**
>
> Osservare le seguenti informazioni.
>
> - AWS AppConfig L'agente viene eseguito sulla porta 2772, per impostazione predefinita. È possibile specificare una porta diversa.
> - È possibile modificare il comportamento predefinito di AWS AppConfig Agent inserendo le variabili di ambiente. Per ulteriori informazioni, consulta Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS.
> - Per *SERVICE_REGION*, specificate il Regione AWS codice (ad esempio`us-west-1`) in cui l' AWS AppConfig agente recupera i dati di configurazione.

3. Eseguite il comando seguente nello `kubectl` strumento per applicare le modifiche al cluster.

```
kubectl apply -f my-deployment.yml
```

4. Al termine della distribuzione, verifica che AWS AppConfig l'agente sia in esecuzione. Utilizzate il comando seguente per visualizzare il file di registro del pod dell'applicazione.

```
kubectl logs -n my-namespace -c appconfig-agent my-pod
```

Individuate un'istruzione come la seguente per il contenitore AWS AppConfig Agent: `[appconfig agent] 1970/01/01 00:00:00 INFO serving on localhost:2772`

> **Note**
>
> È possibile regolare il comportamento predefinito di AWS AppConfig Agent inserendo o modificando le variabili di ambiente. Per informazioni sulle variabili di ambiente disponibili, vedereUtilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS.

Utilizzo di variabili di ambiente per configurare AWS AppConfig Agent for Amazon ECS e Amazon EKS

Puoi configurare AWS AppConfig Agent modificando le seguenti variabili di ambiente per il contenitore dell'agente.

Variabile di ambiente	Informazioni	Valore predefinito

`ACCESS_TOKEN`	Questa variabile di ambiente definisce un token che deve essere fornito quando si richiedono i dati di configura zione dal server HTTP dell'agente. Il valore del token deve essere impostato nell'inte stazione di autorizzazione della richiesta HTTP con un tipo di autorizzazione di. `Bearer` Ecco un esempio.
	`GET /applications/my_a pp/... Host: localhost:2772`
	`Authorization: Bearer <token value>`
		Nessuno

`BACKUP_DIRECTORY`	Questa variabile di ambiente consente all' AWS AppConfig	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
	agente di salvare un backup di ogni configurazione recuperat a nella directory specificata. **Important** Le configurazioni di cui è stato eseguito il backup su disco non sono crittogra fate. Se la configura zione contiene dati sensibili, si AWS AppConfig consiglia di applicare il principio del privilegio minimo con le autorizzazioni del file system. Per ulteriori informazioni, consulta Sicurezza in AWS AppConfig.	
`HTTP_PORT`	Questa variabile di ambiente specifica la porta su cui viene eseguito il server HTTP per l'agente.	2772

Variabile di ambiente	Informazioni	Valore predefinito
`LOG_LEVEL`	Questa variabile di ambiente specifica il livello di dettaglio registrato dall'agente. Ogni livello include il livello corrente e tutti i livelli superiori. Le variabili distinguono tra maiuscole e minuscole. Dal più dettaglia to al meno dettagliato, i livelli di registro sono: `debug`,`info`,`warn`,`error`, e`none`. `Debug`include informazioni dettagliate, incluse informazioni sulla tempistica, sull'agente.	`info`

Variabile di ambiente	Informazioni	Valore predefinito
`MANIFEST`	Questa variabile di ambiente configura l' AWS AppConfig agente per sfruttare funzionalità aggiuntive relative alla configurazione, come il recupero di più account e il salvataggio della configurazione su disco. È possibile inserire uno dei seguenti valori: • `"app:env:manifest-config"` • `"file:/fully/qualified/path/to/manifest.json"` Per ulteriori informazioni su queste caratteristiche, consultare Funzionalità di recupero aggiuntive.	true
`MAX_CONNECTIONS`	Questa variabile di ambiente configura il numero massimo di connessioni utilizzate dall'agente per recuperare le configurazioni. AWS AppConfig	3

Variabile di ambiente	Informazioni	Valore predefinito
`POLL_INTERVAL`	Questa variabile di ambiente controlla la frequenza con cui l'agente richiede dati di configurazione aggiornati. AWS AppConfig È possibile specificare un numero di secondi per l'intervallo. È inoltre possibile specificare un numero con un'unità di tempo: s per secondi, m per minuti e h per ore. Se non viene specificata un'unità, l'agente utilizza come impostazi one predefinita i secondi. Ad esempio, 60, 60 e 1 m generano lo stesso intervallo di sondaggio.	45 secondi
`PREFETCH_LIST`	Questa variabile di ambiente specifica i dati di configura zione richiesti dall'agente non AWS AppConfig appena viene avviato.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
PRELOAD_BACKUPS	Se impostato su`true`, AWS AppConfig l'agente carica i backup di configura zione presenti BACKUP_DI RECTORY nella memoria e verifica immediatamente se esiste una versione più recente del servizio. Se impostato su`false`, l' AWS AppConfig agente carica i contenuti da un backup di configurazione solo se non è in grado di recuperare i dati di configurazione dal servizio, ad esempio se c'è un problema con la rete.	true
PROXY_HEADERS	Questa variabile di ambiente specifica le intestazioni richieste dal proxy a cui fa riferimento la variabile di ambiente. PROXY_URL Il valore è un elenco di intestazi oni separate da virgole. Ogni intestazione utilizza il seguente modulo. `"header: value"`	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
PROXY_URL	Questa variabile di ambiente specifica l'URL del proxy da utilizzare per le connessioni dall'agente a Servizi AWS, incluso. AWS AppConfig HTTPSe gli HTTP URL sono supportati.	Nessuno

Variabile di ambiente	Informazioni	Valore predefinito
`REQUEST_TIMEOUT`	Questa variabile di ambiente controlla la quantità di tempo da cui l'agente attende una risposta. AWS AppConfig Se il servizio non risponde, la richiesta ha esito negativo. Se la richiesta riguarda il recupero iniziale dei dati, l'agente restituisce un errore all'applicazione. Se il timeout si verifica durante un controllo in background per verificare la presenza di dati aggiornati, l'agente registra l'errore e riprova dopo un breve ritardo. È possibile specificare il numero di millisecondi per il timeout. È inoltre possibile specificare un numero con un'unità di tempo: ms per millisecondi e s per secondi. Se non viene specificata un'unità, l'agente utilizza come impostazione predefinita i millisecondi. Ad esempio, 5000, 5000 ms e 5 secondi generano lo stesso valore di timeout della richiesta.	3000 millisecondi

Variabile di ambiente	Informazioni	Valore predefinito
ROLE_ARN	Questa variabile di ambiente specifica l'Amazon Resource Name (ARN) di un ruolo IAM. AWS AppConfig L'agente assume questo ruolo per recuperare i dati di configura zione.	Nessuno
ROLE_EXTERNAL_ID	Questa variabile di ambiente specifica l'ID esterno da utilizzare con il ruolo ARN assunto.	Nessuno
ROLE_SESSION_NAME	Questa variabile di ambiente specifica il nome della sessione da associare alle credenziali per il ruolo IAM assunto.	Nessuno
SERVICE_REGION	Questa variabile di ambiente specifica un'alternativa Regione AWS utilizzata da AWS AppConfig Agent per chiamare il servizio. AWS AppConfig Se non viene definita, l'agente tenta di determinare la regione corrente. In caso contrario, l'agente non si avvia.	Nessuno
WAIT_ON_MANIFEST	Questa variabile di ambiente configura l' AWS AppConfig agente in modo che attenda l'elaborazione del manifesto prima di completare l'avvio.	true

Recupero dei dati di configurazione

È possibile recuperare i dati di configurazione dall' AWS AppConfig agente utilizzando una chiamata HTTP localhost. Gli esempi seguenti vengono utilizzati `curl` con un client HTTP. È possibile chiamare l'agente utilizzando qualsiasi client HTTP disponibile supportato dal linguaggio dell'applicazione o dalle librerie disponibili.

> **Note**
>
> Per recuperare i dati di configurazione se l'applicazione utilizza una barra, ad esempio «test-backend/test-service», è necessario utilizzare la codifica URL.

Per recuperare il contenuto completo di qualsiasi configurazione distribuita

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name"
```

Per recuperare un singolo flag e i relativi attributi da una AWS AppConfig configurazione di tipo **`Feature Flag`**

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?flag=flag_name"
```

Per accedere a più flag e ai relativi attributi da una AWS AppConfig configurazione di tipo **`Feature Flag`**

```
$ curl "http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name?
flag=flag_name_one&flag=flag_name_two"
```

Funzionalità di recupero aggiuntive

AWS AppConfig Agent offre le seguenti funzionalità aggiuntive per aiutarvi a recuperare le configurazioni per le vostre applicazioni.

- Recupero di più account: Utilizzate AWS AppConfig Agent da un sistema primario o di recupero per recuperare i dati di configurazione Account AWS da account di più fornitori.

- Scrivi una copia della configurazione su disco: utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.

Informazioni sui manifesti degli agenti

Per abilitare queste funzionalità AWS AppConfig dell'agente, è necessario creare un manifesto. Un manifesto è un insieme di dati di configurazione forniti per controllare le azioni che l'agente può eseguire. Un manifesto è scritto in JSON. Contiene un set di chiavi di primo livello che corrispondono a diverse configurazioni che hai distribuito utilizzando. AWS AppConfig

Un manifesto può includere più configurazioni. Inoltre, ogni configurazione nel manifesto può identificare una o più funzionalità dell'agente da utilizzare per la configurazione specificata. Il contenuto del manifesto utilizza il seguente formato:

```
{
    "application_name:environment_name:configuration_name": {
        "agent_feature_to_enable_1": {
            "feature-setting-key": "feature-setting-value"
        },
        "agent_feature_to_enable_2": {
            "feature-setting-key": "feature-setting-value"
        }
    }
}
```

Ecco un esempio di JSON per un manifesto con due configurazioni. La prima configurazione (*MyApp*) non utilizza alcuna funzionalità AWS AppConfig dell'agente. La seconda configurazione (*My2ndApp*) utilizza la copia di scrittura della configurazione su disco e le funzionalità di recupero di più account:

```
{
        "MyApp:Test:MyAllowListConfiguration": {},

        "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {
            "credentials": {
                "roleArn": "arn:us-west-1:iam::123456789012:role/MyTestRole",
                "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",
                "roleSessionName": "AwsAppConfigAgent",
                "credentialsDuration": "2h"
            },
```

```
            "writeTo": {
                "path": "/tmp/aws-appconfig/my-2nd-app/beta/my-enable-payments-feature-
flag-configuration.json"
            }
        }
    }
```

Come fornire un manifesto dell'agente

È possibile archiviare il manifesto come file in una posizione in cui AWS AppConfig Agent può leggerlo. In alternativa, è possibile archiviare il manifesto come AWS AppConfig configurazione e indirizzare l'agente verso di esso. Per fornire un manifesto dell'agente, è necessario impostare una variabile di `MANIFEST` ambiente con uno dei seguenti valori:

Posizione del manifesto	Valore della variabile d'ambiente	Caso d'uso
File	file: /path/to/agent-manifest.json	Usa questo metodo se il tuo manifest non cambia spesso.
AWS AppConfig configurazione	*nome-applicazione: nome-ambiente: nome-configurazione*	Utilizzate questo metodo per gli aggiornamenti dinamici. È possibile aggiornare e distribuire un manifesto archiviato AWS AppConfig come configurazione nello stesso modo in cui si archiviano altre AWS AppConfig configurazioni.
Variabile di ambiente	Contenuto del manifesto (JSON)	Usa questo metodo se il tuo manifesto non cambia spesso. Questo metodo è utile in ambienti container in cui è più facile impostare una variabile di ambiente piuttosto che esporre un file.

Per ulteriori informazioni sull'impostazione delle variabili per AWS AppConfig Agent, consultate l'argomento pertinente al vostro caso d'uso:

- Configurazione dell'estensione AWS AppConfig Agent Lambda
- Utilizzo di AWS AppConfig Agent con Amazon EC2
- Utilizzo di AWS AppConfig Agent con Amazon ECS e Amazon EKS

Recupero di più account

È possibile configurare AWS AppConfig l'agente per recuperare le configurazioni da più configurazioni Account AWS inserendo le sostituzioni delle credenziali nel manifesto dell'agente. AWS AppConfig Le sostituzioni delle credenziali includono l'Amazon Resource Name (ARN) di un ruolo AWS Identity and Access Management (IAM), un ID di ruolo, un nome di sessione e la durata per cui l'agente può assumere il ruolo.

Inserisci questi dettagli in una sezione «credenziali» del manifesto. La sezione «credenziali» utilizza il seguente formato:

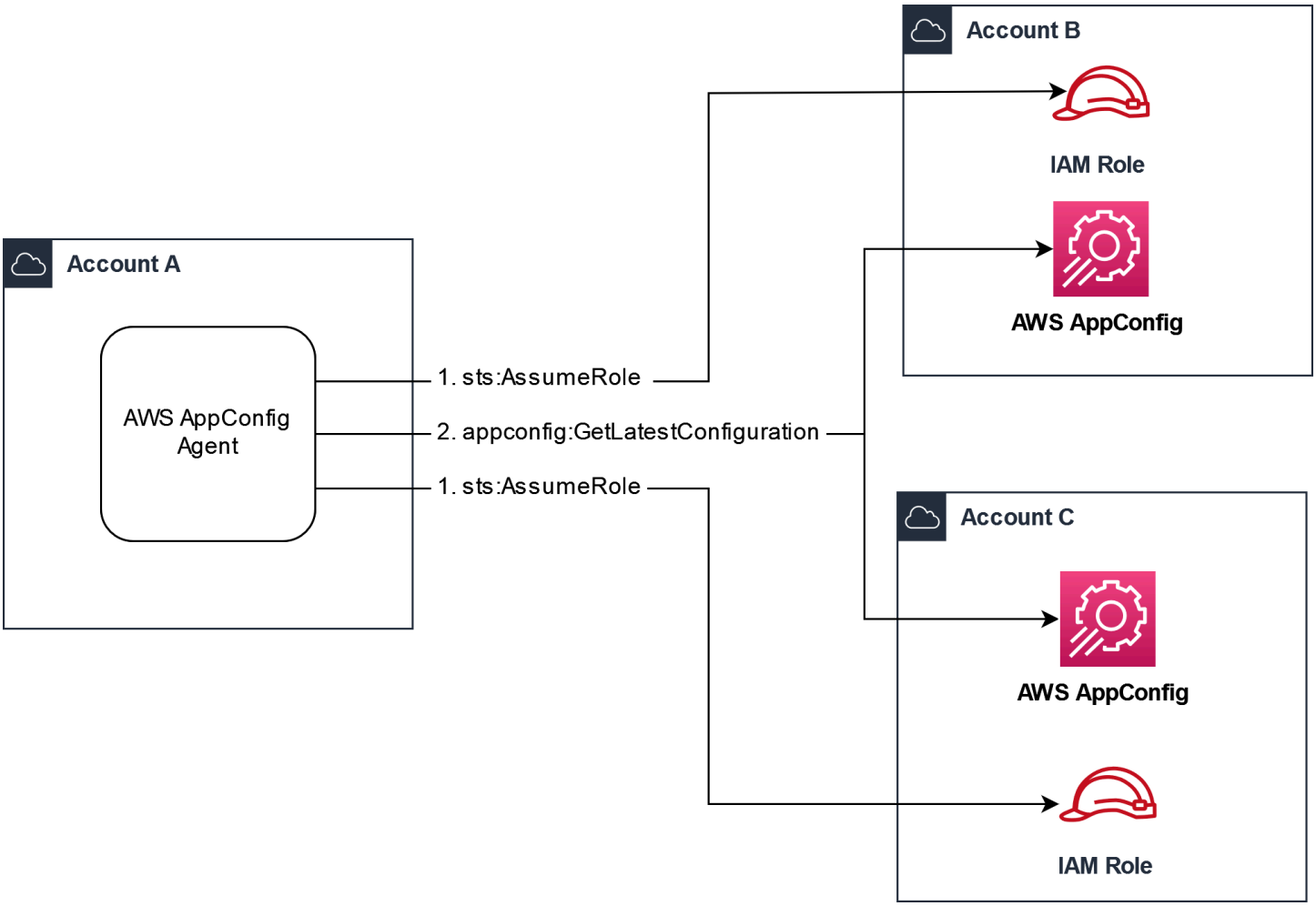
```
{
    "application_name:environment_name:configuration_name": {
        "credentials": {
            "roleArn": "arn:partition:iam::account_ID:role/roleName",
            "roleExternalId": "string",
            "roleSessionName": "string",
            "credentialsDuration": "time_in_hours"
        }
    }
}
```

Ecco un esempio:

```
{
    "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {
        "credentials": {
            "roleArn": "arn:us-west-1:iam::123456789012:role/MyTestRole",
            "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",
            "roleSessionName": "AWSAppConfigAgent",
            "credentialsDuration": "2h"
        }
    }
```

```
}
```

Prima di recuperare una configurazione, l'agente legge i dettagli delle credenziali per la configurazione dal manifest e quindi assume il ruolo IAM specificato per quella configurazione. È possibile specificare un set diverso di sostituzioni di credenziali per diverse configurazioni in un unico manifesto. Il diagramma seguente mostra come l' AWS AppConfig agente, durante l'esecuzione nell'account A (l'account di recupero), assuma ruoli separati specificati per gli account B e C (gli account fornitore) e quindi richiami l'operazione GetLatestConfigurationAPI per recuperare i dati di configurazione dall'esecuzione in tali account: AWS AppConfig



Configura le autorizzazioni per recuperare i dati di configurazione dagli account dei fornitori

AWS AppConfig L'agente in esecuzione nell'account di recupero necessita dell'autorizzazione per recuperare i dati di configurazione dagli account del fornitore. Si concede l'autorizzazione all'agente creando un ruolo AWS Identity and Access Management (IAM) in ciascuno degli account del fornitore. AWS AppConfig L'agente nell'account di recupero assume questo ruolo per ottenere

dati dagli account dei fornitori. Completa le procedure in questa sezione per creare una policy di autorizzazioni IAM, un ruolo IAM e aggiungere le sostituzioni degli agenti al manifesto.

Prima di iniziare

Raccogli le seguenti informazioni prima di creare una politica di autorizzazione e un ruolo in IAM.

- Gli ID per ciascuno Account AWS. L'account di recupero è l'account che chiamerà altri account per i dati di configurazione. Gli account fornitore sono gli account che invieranno i dati di configurazione all'account di recupero.
- Il nome del ruolo IAM utilizzato da AWS AppConfig nell'account di recupero. Ecco un elenco dei ruoli utilizzati da AWS AppConfig, per impostazione predefinita:
  - Per Amazon Elastic Compute Cloud (Amazon EC2) AWS AppConfig , utilizza il ruolo di istanza.
  - For AWS Lambda, AWS AppConfig utilizza il ruolo di esecuzione Lambda.
  - Per Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS), utilizza il ruolo contenitore. AWS AppConfig

  Se hai configurato l' AWS AppConfig agente per utilizzare un ruolo IAM diverso specificando la variabile di `ROLE_ARN` ambiente, prendi nota di quel nome.

Crea la politica delle autorizzazioni

Utilizza la seguente procedura per creare una politica di autorizzazioni utilizzando la console IAM. Completa la procedura in ciascuna unità Account AWS che fornirà i dati di configurazione per l'account di recupero.

Per creare una policy IAM

1. Accedi all'account di un AWS Management Console fornitore.
2. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.
4. Scegli l'opzione JSON.
5. Nell'editor delle politiche, sostituisci il codice JSON predefinito con la seguente dichiarazione di policy. Aggiorna ogni *`segnaposto di risorse di esempio`* con i dettagli dell'account del fornitore.

   ```
   {
   ```

```
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "appconfig:StartConfigurationSession",
            "appconfig:GetLatestConfiguration"
        ],
        "Resource":
 "arn:partition:appconfig:region:vendor_account_ID:application/
vendor_application_ID/environment/vendor_environment_ID/
configuration/vendor_configuration_ID"
    }
   ]
}
```

Ecco un esempio:

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "appconfig:StartConfigurationSession",
            "appconfig:GetLatestConfiguration"
        ],
        "Resource": "arn:aws:appconfig:us-east-2:111122223333:application/abc123/
environment/def456/configuration/hij789"
    }
   ]
}
```

6. Seleziona Successivo.
7. Nel campo Nome della politica, inserisci un nome.
8. (Facoltativo) Per Aggiungi tag, aggiungi una o più coppie tag-chiave-valore per organizzare, tracciare o controllare l'accesso a questa politica.
9. Scegli Crea policy. Il sistema visualizza di nuovo la pagina Policies (Policy).
10. Ripetete questa procedura in ciascuna unità Account AWS che fornirà i dati di configurazione per l'account di recupero.

Crea il ruolo IAM

Utilizza la seguente procedura per creare un ruolo IAM utilizzando la console IAM. Completa la procedura in ciascuna Account AWS unità che fornirà i dati di configurazione per l'account di recupero.

Per creare un ruolo IAM

1. Accedi all'account di un AWS Management Console fornitore.
2. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
3. Nel riquadro di navigazione, scegli Ruoli, quindi scegli Crea politica.
4. Per Trusted entity type (Tipo di entità attendibile), scegli Account AWS.
5. Nella Account AWSsezione, scegli Altro Account AWS.
6. Nel campo ID account, inserisci l'ID dell'account di recupero.
7. (Facoltativo) Come procedura consigliata di sicurezza per questo ruolo, scegli Richiedi ID esterno e inserisci una stringa.
8. Seleziona Successivo.
9. Nella pagina Aggiungi autorizzazioni, utilizza il campo Cerca per individuare la politica creata nella procedura precedente. Seleziona la casella accanto al suo nome.
10. Seleziona Successivo.
11. In Role name (Nome ruolo), immettere un nome.
12. (Facoltativo) In Description (Descrizione), immettere una descrizione.
13. Per il passaggio 1: seleziona le entità attendibili, scegli Modifica. Sostituisci la politica di fiducia JSON predefinita con la seguente politica. Aggiorna ogni *segnaposto di risorse di esempio* con le informazioni del tuo account di recupero.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS":
 "arn:aws:iam::retrieval_account_ID:role/appconfig_role_in_retrieval_account"
            },
            "Action": "sts:AssumeRole"
        }
    ]
```

```
}
```

14. (Facoltativo) In Tags (Tag), aggiungi una o più coppie tag chiave-valore per organizzare, monitorare o controllare l'accesso per questo ruolo.
15. Scegliere Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Roles (Ruoli).
16. Cerca il ruolo che hai appena creato. Sceglilo. Nella sezione ARN, copia l'ARN. Queste informazioni verranno specificate nella procedura successiva.

Aggiungi sostituzioni di credenziali al manifesto

Dopo aver creato il ruolo IAM nel tuo account fornitore, aggiorna il manifesto nell'account di recupero. In particolare, aggiungi il blocco delle credenziali e l'ARN del ruolo IAM per recuperare i dati di configurazione dall'account del fornitore. Ecco il formato JSON:

```
{
    "vendor_application_name:vendor_environment_name:vendor_configuration_name": {
        "credentials": {
            "roleArn":
 "arn:partition:iam::vendor_account_ID:role/name_of_role_created_in_vendor_account",
            "roleExternalId": "string",
            "roleSessionName": "string",
            "credentialsDuration": "time_in_hours"
        }
    }
}
```

Ecco un esempio:

```
{
    "My2ndApp:Beta:MyEnableMobilePaymentsFeatureFlagConfiguration": {
        "credentials": {
            "roleArn": "arn:us-west-1:iam::123456789012:role/MyTestRole",
            "roleExternalId": "00b148e2-4ea4-46a1-ab0f-c422b54d0aac",
            "roleSessionName": "AwsAppConfigAgent",
            "credentialsDuration": "2h"
        }
    }
}
```

Verifica che il recupero da più account funzioni

È possibile verificare che l'agente sia in grado di recuperare i dati di configurazione da più account esaminando i registri dell'agente. AWS AppConfig Il registro dei `INFO` livelli per i dati iniziali recuperati per '`YourApplicationName:YourEnvironmentName:YourConfigurationName`' è l'indicatore migliore per il successo dei recuperi. Se i recuperi non riescono, dovrebbe apparire un registro dei `ERROR` livelli che indica il motivo dell'errore. Di seguito è riportato un esempio di recupero riuscito da un account fornitore:

```
[appconfig agent] 2023/11/13 11:33:27 INFO AppConfig Agent 2.0.x
[appconfig agent] 2023/11/13 11:33:28 INFO serving on localhost:2772
[appconfig agent] 2023/11/13 11:33:28 INFO retrieved initial data for
 'MyTestApplication:MyTestEnvironment:MyDenyListConfiguration' in XX.Xms
```

Scrivi una copia della configurazione su disco

È possibile configurare AWS AppConfig Agent per archiviare automaticamente una copia di una configurazione su disco in formato testo normale. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig.

Questa funzionalità non è progettata per essere utilizzata come funzionalità di backup della configurazione. AWS AppConfig L'agente non legge i file di configurazione copiati su disco. Se desideri eseguire il backup delle configurazioni su disco, consulta le `BACKUP_DIRECTORY` variabili di `PRELOAD_BACKUP` ambiente per Using AWS AppConfig Agent with Amazon EC2 o AWS AppConfig Using Agent with Amazon ECS e Amazon EKS.

> **Warning**
>
> Tieni presente le seguenti informazioni importanti su questa funzionalità:
>
> - Le configurazioni salvate su disco vengono archiviate in testo semplice e sono leggibili dall'uomo. Non abilitare questa funzionalità per configurazioni che includono dati sensibili.
> - Questa funzionalità scrive sul disco locale. Utilizza il principio del privilegio minimo per le autorizzazioni del filesystem. Per ulteriori informazioni, consulta Implementazione dell'accesso con privilegi minimi.

Per abilitare la scrittura (copia della configurazione su disco)

1. Modifica il manifesto.

2. Scegliete la configurazione che desiderate AWS AppConfig scrivere su disco e aggiungete un `writeTo` elemento. Ecco un esempio:

```
{
    "application_name:environment_name:configuration_name": {
        "writeTo": {
            "path": "path_to_configuration_file"
        }
    }
}
```

   Ecco un esempio:

```
{
    "MyTestApp:MyTestEnvironment:MyNewConfiguration": {
        "writeTo": {
            "path": "/tmp/aws-appconfig/mobile-app/beta/enable-mobile-payments"
        }
    }
}
```

3. Salvare le modifiche. Il file configuration.json verrà aggiornato ogni volta che vengono distribuiti nuovi dati di configurazione.

Verifica che la copia di scrittura della configurazione su disco funzioni

È possibile verificare che le copie di una configurazione vengano scritte su disco esaminando i log dell' AWS AppConfig agente. La voce di `INFO` registro con la frase «INFO write configuration '*application*: *environment*: *configuration*' to *file_path*" indica che l' AWS AppConfig agente scrive copie di configurazione su disco.

Ecco un esempio:

```
[appconfig agent] 2023/11/13 11:33:27 INFO AppConfig Agent 2.0.x
[appconfig agent] 2023/11/13 11:33:28 INFO serving on localhost:2772
[appconfig agent] 2023/11/13 11:33:28 INFO retrieved initial data for
 'MobileApp:Beta:EnableMobilePayments' in XX.Xms
[appconfig agent] 2023/11/13 17:05:49 INFO wrote configuration
 'MobileApp:Beta:EnableMobilePayments' to /tmp/configs/your-app/your-env/your-
config.json
```

AWS AppConfig Sviluppo locale dell'agente

AWS AppConfig L'agente supporta una modalità di sviluppo locale. Se si abilita la modalità di sviluppo locale, l'agente legge i dati di configurazione da una directory specificata su disco. Non recupera i dati di configurazione da. AWS AppConfigÈ possibile simulare le distribuzioni di configurazione aggiornando i file nella directory specificata. Consigliamo la modalità di sviluppo locale per i seguenti casi d'uso:

- Prova diverse versioni di configurazione prima di distribuirle utilizzando AWS AppConfig.
- Prova diverse opzioni di configurazione per una nuova funzionalità prima di apportare modifiche al tuo repository di codice.
- Prova diversi scenari di configurazione per verificare che funzionino come previsto.

> **Warning**
>
> Non utilizzate la modalità di sviluppo locale negli ambienti di produzione. Questa modalità non supporta importanti funzionalità di AWS AppConfig sicurezza come la convalida dell'implementazione e i rollback automatici.

Utilizzare la procedura seguente per configurare AWS AppConfig Agent per la modalità di sviluppo locale.

Per configurare AWS AppConfig Agent per la modalità di sviluppo locale

1. Installa l'agente utilizzando il metodo descritto per il tuo ambiente di calcolo. AWS AppConfig L'agente funziona con quanto segue Servizi AWS:
   - AWS Lambda
   - Amazon EC2
   - Amazon ECS e Amazon EKS
2. Se l'agente è in esecuzione, interrompilo.
3. Aggiungi `LOCAL_DEVELOPMENT_DIRECTORY` all'elenco delle variabili di ambiente. Specificate una directory sul filesystem che fornisca all'agente i permessi di lettura. Ad esempio, `/tmp/local_configs`.
4. Crea un file nella directory. Il nome del file deve utilizzare il seguente formato:

```
application_name:environment_name:configuration_profile_name
```

Ecco un esempio:

```
Mobile:Development:EnableMobilePaymentsFeatureFlagConfiguration
```

> **Note**
>
> (Facoltativo) È possibile controllare il tipo di contenuto restituito dall'agente per i dati di configurazione in base all'estensione assegnata al file. Ad esempio, se denominate il file con un'estensione.json, l'agente restituisce un tipo di contenuto `application/json` quando l'applicazione lo richiede. Se omettete l'estensione, l'agente la utilizza `application/octet-stream` per il tipo di contenuto. Se hai bisogno di un controllo preciso, puoi fornire un'estensione nel formato `.type%subtype`. L'agente restituirà un tipo di contenuto di `.type/subtype`.

5. Esegui il comando seguente per riavviare l'agente e richiedere i dati di configurazione.

```
curl http://localhost:2772/applications/application_name/
environments/environment_name/configurations/configuration_name
```

L'agente verifica le modifiche al file locale all'intervallo di polling specificato per l'agente. Se l'intervallo di sondaggio non è specificato, l'agente utilizza l'intervallo predefinito di 45 secondi. Questo controllo a intervalli di sondaggio assicura che l'agente si comporti in un ambiente di sviluppo locale allo stesso modo in cui si comporta quando è configurato per interagire con il servizio. AWS AppConfig

> **Note**
>
> Per distribuire una nuova versione di un file di configurazione di sviluppo locale, aggiorna il file con nuovi dati.

Recupero delle configurazioni chiamando direttamente le API

L'applicazione recupera i dati di configurazione stabilendo prima una sessione di configurazione utilizzando l'operazione API. StartConfigurationSession Il client della sessione effettua quindi chiamate periodiche per GetLatestConfigurationverificare e recuperare i dati più recenti disponibili.

Durante la chiamata`StartConfigurationSession`, il codice invia le seguenti informazioni:

- Identificatori (ID o nome) di un' AWS AppConfig applicazione, di un ambiente e di un profilo di configurazione monitorati dalla sessione.
- (Facoltativo) La quantità minima di tempo che il client della sessione deve attendere tra una chiamata e l'altra. `GetLatestConfiguration`

In risposta, AWS AppConfig fornisce un `InitialConfigurationToken` messaggio da fornire al client della sessione e da utilizzare la prima volta che effettua una chiamata `GetLatestConfiguration` per quella sessione.

> **Important**
>
> Questo token deve essere usato solo una volta nella prima chiamata a`GetLatestConfiguration`. È necessario utilizzare il nuovo token nella `GetLatestConfiguration` risposta (`NextPollConfigurationToken`) in ogni chiamata successiva a`GetLatestConfiguration`. Per supportare casi d'uso prolungati con sondaggi, i token sono validi per un massimo di 24 ore. Se una `GetLatestConfiguration` chiamata utilizza un token scaduto, il sistema ritorna. `BadRequestException`

Durante la chiamata`GetLatestConfiguration`, il codice cliente invia il `ConfigurationToken` valore più recente a sua disposizione e riceve in risposta:

- `NextPollConfigurationToken`: il `ConfigurationToken` valore da utilizzare nella chiamata successiva a`GetLatestConfiguration`.
- `NextPollIntervalInSeconds`: la durata che il client deve attendere prima di effettuare la chiamata successiva`GetLatestConfiguration`.
- La configurazione: i dati più recenti destinati alla sessione. Questo campo può essere vuoto se il client dispone già dell'ultima versione della configurazione.

> **Important**
>
> Prendi nota delle seguenti informazioni importanti.
>
> - L'StartConfigurationSessionAPI deve essere chiamata solo una volta per applicazione, ambiente, profilo di configurazione e client per stabilire una sessione con il servizio. Questa operazione viene in genere eseguita all'avvio dell'applicazione o immediatamente prima del primo recupero di una configurazione.
> - Se la configurazione viene distribuita utilizzando a`KmsKeyIdentifier`, la richiesta di ricezione della configurazione deve includere l'autorizzazione alla chiamata. `kms:Decrypt` Per ulteriori informazioni, consulta Decrypt nel riferimento API.AWS Key Management Service
> - L'operazione API precedentemente utilizzata per recuperare i dati di configurazione`GetConfiguration`, è obsoleta. L'operazione `GetConfiguration` API non supporta configurazioni crittografate.

Recupero di un esempio di configurazione

L' AWS CLI esempio seguente mostra come recuperare i dati di configurazione utilizzando le AWS AppConfig operazioni Data e API. `StartConfigurationSession` `GetLatestConfiguration` Il primo comando avvia una sessione di configurazione. Questa chiamata include gli ID (o i nomi) dell' AWS AppConfig applicazione, dell'ambiente e del profilo di configurazione. L'API restituisce un file `InitialConfigurationToken` usato per recuperare i dati di configurazione.

```
aws appconfigdata start-configuration-session \
    --application-identifier application_name_or_ID \
    --environment-identifier environment_name_or_ID \
    --configuration-profile-identifier configuration_profile_name_or_ID
```

Il sistema risponde con informazioni nel formato seguente.

```
{
   "InitialConfigurationToken": initial configuration token
}
```

Dopo aver avviato una sessione, usa InitialConfigurationTokento call GetLatestConfigurationper recuperare i dati di configurazione. I dati di configurazione vengono salvati nel `mydata.json` file.

```
aws appconfigdata get-latest-configuration \
    --configuration-token initial configuration token mydata.json
```

La prima chiamata `GetLatestConfiguration` utilizza il codice `ConfigurationToken` ottenuto da`StartConfigurationSession`. Vengono restituite le seguenti informazioni.

```
{
    "NextPollConfigurationToken" : next configuration token,
    "ContentType" : content type of configuration,
    "NextPollIntervalInSeconds" : 60
}
```

Le chiamate successive `GetLatestConfiguration` devono `NextPollConfigurationToken` provenire dalla risposta precedente.

```
aws appconfigdata get-latest-configuration \
    --configuration-token next configuration token mydata.json
```

Important

Tieni presente i seguenti dettagli importanti sul funzionamento dell'`GetLatestConfiguration`API:

- La `GetLatestConfiguration` risposta include una `Configuration` sezione che mostra i dati di configurazione. La `Configuration` sezione viene visualizzata solo se il sistema trova dati di configurazione nuovi o aggiornati. Se il sistema non trova dati di configurazione nuovi o aggiornati, i `Configuration` dati sono vuoti.
- Ne ricevi uno nuovo `ConfigurationToken` in ogni risposta da`GetLatestConfiguration`.
- Si consiglia di ottimizzare la frequenza di polling delle chiamate API `GetLatestConfiguration` in base al budget, alla frequenza prevista delle distribuzioni di configurazione e al numero di destinazioni per una configurazione.

Estendere i flussi di lavoro utilizzando le estensioni

Un'estensione aumenta la capacità di inserire logica o comportamento in punti diversi durante il AWS AppConfig flusso di lavoro di creazione o distribuzione di una configurazione. Ad esempio, puoi utilizzare le estensioni per eseguire i seguenti tipi di attività (solo per citarne alcune):

- Invia una notifica a un argomento di Amazon Simple Notification Service (Amazon SNS) quando viene distribuito un profilo di configurazione.
- Scorri il contenuto di un profilo di configurazione alla ricerca di dati sensibili prima dell'inizio della distribuzione.
- Crea o aggiorna un problema di Atlassian Jira ogni volta che viene apportata una modifica a un flag di funzionalità.
- Unisci i contenuti di un servizio o di una fonte di dati ai dati di configurazione quando avvii una distribuzione.
- Esegui il backup di una configurazione in un bucket Amazon Simple Storage Service (Amazon S3) ogni volta che viene distribuita una configurazione.

Puoi associare questi tipi di attività ad AWS AppConfig applicazioni, ambienti e profili di configurazione.

Indice

Informazioni sulle AWS AppConfig estensioni

Questo argomento introduce i concetti e la terminologia relativi alle AWS AppConfig estensioni. Le informazioni vengono discusse nel contesto di ogni passaggio necessario per configurare e utilizzare AWS AppConfig le estensioni.

Argomenti

- Passo 2: Determina quando vuoi che l'estensione venga eseguita
- Passaggio 3: Creare un'associazione di estensioni
- Passaggio 4: Implementa una configurazione e verifica che le azioni di estensione siano state eseguite

Passaggio 1: Stabilisci cosa vuoi fare con le estensioni

Vuoi ricevere una notifica su un webhook che invia messaggi a Slack ogni volta che viene completata una AWS AppConfig distribuzione? Vuoi eseguire il backup di un profilo di configurazione in un bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) prima di implementare una configurazione? Vuoi cancellare i dati di configurazione alla ricerca di informazioni sensibili prima che la configurazione venga distribuita? È possibile utilizzare le estensioni per eseguire questo tipo di attività e altro ancora. È possibile creare estensioni personalizzate o utilizzare le estensioni AWS create incluse in AWS AppConfig.

> **Note**
>
> Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Per ulteriori informazioni, consulta Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig.

Le seguenti estensioni AWS create possono aiutarti a integrare rapidamente le distribuzioni di configurazione con altri servizi. Puoi utilizzare queste estensioni nella AWS AppConfig console o richiamando le azioni dell'API delle estensioni direttamente da AWS CLI AWS Tools for PowerShell, o dall'SDK.

Estensione	Descrizione
Amazon CloudWatch evidentemente sta testando un A/B	Questa estensione consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'operazione. EvaluateFeature Per ulteriori informazioni, consulta Lavorare con l'estensione Amazon CloudWatch Evidently.

Estensione	Descrizione
AWS AppConfig eventi di distribuzione a EventBridge	Questa estensione invia gli eventi al bus degli eventi EventBridge predefinito quando viene distribuita una configurazione.
AWS AppConfig eventi di distribuzione su Amazon Simple Notification Service (Amazon SNS)	Questa estensione invia messaggi a un argomento di Amazon SNS specificato quando viene distribuita una configurazione.
AWS AppConfig eventi di distribuzione su Amazon Simple Queue Service (Amazon SQS)	Questa estensione inserisce i messaggi nella coda di Amazon SQS quando viene distribuita una configurazione.
Estensione di integrazione: Atlassian Jira	Queste estensioni consentono di AWS AppConfig creare e aggiornare problemi ogni volta che si apportano modifiche a un feature flag.

Passo 2: Determina quando vuoi che l'estensione venga eseguita

Un'estensione definisce una o più azioni che esegue durante un AWS AppConfig flusso di lavoro. Ad esempio, l'`AWS AppConfig deployment events to Amazon SNS`estensione AWS creata include un'azione per inviare una notifica a un argomento di Amazon SNS. Ogni azione viene richiamata quando interagisci con AWS AppConfig o quando AWS AppConfig esegui un processo per tuo conto. Questi sono chiamati punti d'azione. AWS AppConfig le estensioni supportano i seguenti punti di azione:

- `PRE_CREATE_HOSTED_CONFIGURATION_VERSION`
- `PRE_START_DEPLOYMENT`
- `ON_DEPLOYMENT_START`
- `ON_DEPLOYMENT_STEP`
- `ON_DEPLOYMENT_BAKING`
- `ON_DEPLOYMENT_COMPLETE`
- `ON_DEPLOYMENT_ROLLED_BACK`

Le azioni di estensione configurate sui punti di `PRE_*` azione vengono applicate dopo la convalida della richiesta, ma AWS AppConfig prima dell'esecuzione dell'attività corrispondente al nome del punto di azione. Queste chiamate di azione vengono elaborate contemporaneamente a una richiesta. Se viene effettuata più di una richiesta, le chiamate alle azioni vengono eseguite in sequenza. Si noti inoltre che i punti di `PRE_*` azione ricevono e possono modificare il contenuto di una configurazione. `PRE_*`i punti di azione possono anche rispondere a un errore e impedire che si verifichi un'azione.

Un'estensione può anche essere eseguita in parallelo a un AWS AppConfig flusso di lavoro utilizzando un punto di `ON_*` azione. `ON_*`i punti di azione vengono richiamati in modo asincrono. `ON_*`i punti di azione non ricevono il contenuto di una configurazione. Se un'estensione riscontra un errore durante un punto di `ON_*` azione, il servizio ignora l'errore e continua il flusso di lavoro.

Passaggio 3: Creare un'associazione di estensioni

Per creare un'estensione o configurare un' AWS estensione creata, si definiscono i punti di azione che richiamano un'estensione quando viene utilizzata una AWS AppConfig risorsa specifica. Ad esempio, puoi scegliere di eseguire l'`AWS AppConfig deployment events to Amazon SNS`estensione e ricevere notifiche su un argomento di Amazon SNS ogni volta che viene avviata una distribuzione di configurazione per un'applicazione specifica. La definizione dei punti di azione che richiamano un'estensione per una AWS AppConfig risorsa specifica viene chiamata associazione di estensioni. Un'associazione di estensione è una relazione specifica tra un'estensione e una AWS AppConfig risorsa, ad esempio un'applicazione o un profilo di configurazione.

Una singola AWS AppConfig applicazione può includere più ambienti e profili di configurazione. Se si associa un'estensione a un'applicazione o a un ambiente, AWS AppConfig richiama l'estensione per tutti i flussi di lavoro relativi all'applicazione o alle risorse dell'ambiente, se applicabile.

Ad esempio, supponiamo di avere un' AWS AppConfig applicazione chiamata MobileApps che include un profilo di configurazione chiamato. AccessList Supponiamo che l' MobileApps applicazione includa ambienti beta, di integrazione e di produzione. Crei un'associazione di estensione per l' AWS estensione di notifica Amazon SNS creata e associ l'estensione all' MobileApps applicazione. L'estensione di notifica di Amazon SNS viene richiamata ogni volta che la configurazione viene distribuita per l'applicazione in uno dei tre ambienti.

> **Note**
>
> Non è necessario creare un'estensione per utilizzare le estensioni AWS create, ma è necessario creare un'associazione di estensioni.

Passaggio 4: Implementa una configurazione e verifica che le azioni di estensione siano state eseguite

Dopo aver creato un'associazione, quando viene creata una configurazione ospitata o viene distribuita una configurazione, AWS AppConfig richiama l'estensione ed esegue le azioni specificate. Quando viene richiamata un'estensione, se il sistema riscontra un errore durante un punto di `PRE-*` azione, AWS AppConfig restituisce informazioni su tale errore.

Lavorare con le AWS estensioni create

AWS AppConfig include le seguenti estensioni AWS create. Queste estensioni possono aiutarti a integrare il AWS AppConfig flusso di lavoro con altri servizi. Puoi utilizzare queste estensioni in AWS Management Console o chiamando le azioni dell'API delle estensioni direttamente da AWS CLI AWS Tools for PowerShell, o dall'SDK.

Estensione	Descrizione
Amazon CloudWatch evidentemente sta testando un A/B	Questa estensione consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'operazione. EvaluateFeature Per ulteriori informazioni, consulta Lavorare con l'estensione Amazon CloudWatch Evidently.
AWS AppConfig eventi di distribuzione a EventBridge	Questa estensione invia gli eventi al bus degli eventi EventBridge predefinito quando viene distribuita una configurazione.
AWS AppConfig eventi di distribuzione su Amazon Simple Notification Service (Amazon SNS)	Questa estensione invia messaggi a un argomento di Amazon SNS specificato quando viene distribuita una configurazione.
AWS AppConfig eventi di distribuzione su Amazon Simple Queue Service (Amazon SQS)	Questa estensione inserisce i messaggi nella coda di Amazon SQS quando viene distribuita una configurazione.
Estensione di integrazione: Atlassian Jira	Queste estensioni consentono di AWS AppConfig creare e aggiornare problemi ogni

Estensione	Descrizione
	volta che si apportano modifiche a un feature flag.

Lavorare con l'estensione Amazon CloudWatch Evidently

Puoi usare Amazon CloudWatch Evidently per convalidare nuove funzionalità in modo sicuro offrendole disponibili a una percentuale specifica di utenti durante il rollout della funzionalità. È possibile monitorare le prestazioni della nuova funzionalità per aiutarti a decidere quando aumentare il traffico verso gli utenti. Ciò consente di ridurre i rischi e identificare le conseguenze non intenzionali prima di avviare completamente la funzionalità. È inoltre possibile condurre esperimenti A/B per prendere decisioni sulla progettazione delle caratteristiche basate su prove e dati.

L' AWS AppConfig estensione per CloudWatch Evidently consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'operazione. EvaluateFeature Una sessione locale mitiga i rischi di latenza e disponibilità associati a una chiamata API. Per informazioni su come configurare e utilizzare l'estensione, consulta Esegui lanci ed esperimenti A/B con CloudWatch Evidently nella Amazon CloudWatch User Guide.

Lavorare con l'estensione **`AWS AppConfig deployment events to Amazon EventBridge`**

L'`AWS AppConfig deployment events to Amazon EventBridge`estensione è un'estensione AWS creata che consente di monitorare e modificare il flusso di lavoro di installazione della AWS AppConfig configurazione. L'estensione invia notifiche di eventi al bus degli eventi EventBridge predefinito ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle AWS AppConfig applicazioni, degli ambienti o dei profili di configurazione, AWS AppConfig invia notifiche di eventi al bus degli eventi dopo l'inizio, la fine e il rollback di ogni implementazione della configurazione.

Se desideri un maggiore controllo sui punti di azione che inviano EventBridge le notifiche, puoi creare un'estensione personalizzata e inserire il bus degli eventi EventBridge predefinito Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consultaProcedura dettagliata: creazione di estensioni personalizzate AWS AppConfig.

> **Important**
>
> Questa estensione supporta solo il bus degli eventi EventBridge predefinito.

Utilizzo dell'estensione

Per utilizzare l'`AWS AppConfig deployment events to Amazon EventBridge`estensione, devi innanzitutto collegarla a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione CreateExtensionAssociationAPI. Quando si crea l'associazione, si specifica l'ARN di un' AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica di evento per qualsiasi profilo di configurazione contenuto nell'applicazione o nell'ambiente specificato.

Dopo aver creato l'associazione, quando viene distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

> **Note**
>
> Questa estensione viene richiamata dai seguenti punti di azione:
>
> - `ON_DEPLOYMENT_START`
> - `ON_DEPLOYMENT_COMPLETE`
> - `ON_DEPLOYMENT_ROLLED_BACK`
>
> Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig.

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel riquadro di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.
5. Scegli Crea associazione alla risorsa.

Ecco un esempio di evento inviato EventBridge quando viene richiamata l'estensione.

```
{
   "version":"0",
   "id":"c53dbd72-c1a0-2302-9ed6-c076e9128277",
   "detail-type":"On Deployment Complete",
   "source":"aws.appconfig",
   "account":"111122223333",
   "time":"2022-07-09T01:44:15Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:appconfig:us-east-1:111122223333:extensionassociation/z763ff5"
   ],
   "detail":{
      "InvocationId":"5tfjcig",
       "Parameters":{

      },
      "Type":"OnDeploymentComplete",
      "Application":{
         "Id":"ba8toh7",
         "Name":"MyApp"
      },
      "Environment":{
         "Id":"pgil2o7",
         "Name":"MyEnv"
      },
      "ConfigurationProfile":{
         "Id":"ga3tqep",
```

```
            "Name":"MyConfigProfile"
        },
        "DeploymentNumber":1,
        "ConfigurationVersion":"1"
    }
}
```

Lavorare con l'estensione **AWS AppConfig deployment events to Amazon SNS**

L'`AWS AppConfig deployment events to Amazon SNS`estensione è un'estensione AWS creata che consente di monitorare e modificare il flusso di lavoro di installazione della AWS AppConfig configurazione. L'estensione pubblica messaggi su un argomento di Amazon SNS ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle tue AWS AppConfig applicazioni, ambienti o profili di configurazione, AWS AppConfig pubblica un messaggio sull'argomento dopo ogni inizio, fine e ripristino della configurazione.

Se desideri un maggiore controllo su quali punti di azione inviano notifiche Amazon SNS, puoi creare un'estensione personalizzata e inserire un argomento Amazon SNS Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consulta. Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig

Utilizzo dell'estensione

Questa sezione descrive come utilizzare l'`AWS AppConfig deployment events to Amazon SNS`estensione.

Fase 1: AWS AppConfig Configurare la pubblicazione dei messaggi su un argomento

Aggiungi una policy di controllo degli accessi al tuo argomento Amazon SNS: concessione delle autorizzazioni di pubblicazione AWS AppConfig (`appconfig.amazonaws.com`). `sns:Publish` Per ulteriori informazioni, consulta Casi di esempio per il controllo degli accessi di Amazon SNS.

Passaggio 2: creare un'associazione di estensioni

Collega l'estensione a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione CreateExtensionAssociationAPI. Quando si crea l'associazione, si specifica l'ARN di un' AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica per qualsiasi profilo di configurazione

contenuto nell'applicazione o nell'ambiente specificato. Quando crei l'associazione, devi inserire un valore per il `topicArn` parametro che contiene l'ARN dell'argomento Amazon SNS che desideri utilizzare.

Dopo aver creato l'associazione, quando viene distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

> **Note**
>
> Questa estensione viene richiamata dai seguenti punti di azione:
>
> - `ON_DEPLOYMENT_START`
> - `ON_DEPLOYMENT_COMPLETE`
> - `ON_DEPLOYMENT_ROLLED_BACK`
>
> Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig.

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel riquadro di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.
5. Scegli Crea associazione alla risorsa.

Ecco un esempio del messaggio inviato all'argomento Amazon SNS quando viene richiamata l'estensione.

```
{
    "Type": "Notification",
    "MessageId": "ae9d702f-9a66-51b3-8586-2b17932a9f28",
    "TopicArn": "arn:aws:sns:us-east-1:111122223333:MySNSTopic",
    "Message": {
        "InvocationId": "7itcaxp",
        "Parameters": {
            "topicArn": "arn:aws:sns:us-east-1:111122223333:MySNSTopic"
        },
        "Application": {
            "Id": "1a2b3c4d",
            "Name": MyApp
        },
        "Environment": {
            "Id": "1a2b3c4d",
            "Name": MyEnv
        },
        "ConfigurationProfile": {
            "Id": "1a2b3c4d",
            "Name": "MyConfigProfile"
        },
        "Description": null,
        "DeploymentNumber": "3",
        "ConfigurationVersion": "1",
        "Type": "OnDeploymentComplete"
    },
    "Timestamp": "2022-06-30T20:26:52.067Z",
    "SignatureVersion": "1",
    "Signature": "<...>",
    "SigningCertURL": "<...>",
    "UnsubscribeURL": "<...>",
    "MessageAttributes": {
        "MessageType": {
            "Type": "String",
            "Value": "OnDeploymentStart"
        }
    }
}
```

Lavorare con l'estensione **AWS AppConfig deployment events to Amazon SQS**

L'`AWS AppConfig deployment events to Amazon SQS`estensione è un'estensione AWS creata che consente di monitorare e modificare il flusso di lavoro di installazione della AWS AppConfig configurazione. L'estensione inserisce i messaggi nella coda di Amazon Simple Queue Service (Amazon SQS) ogni volta che viene distribuita una configurazione. Dopo aver associato l'estensione a una delle tue AWS AppConfig applicazioni, ambienti o profili di configurazione, AWS AppConfig inserisce un messaggio nella coda dopo ogni inizio, fine e rollback della configurazione.

Se desideri un maggiore controllo su quali punti di azione inviano notifiche Amazon SQS, puoi creare un'estensione personalizzata e inserire una coda Amazon SQS Amazon Resource Name (ARN) per il campo URI. Per informazioni sulla creazione di un'estensione, consulta. Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig

Utilizzo dell'estensione

Questa sezione descrive come utilizzare l'`AWS AppConfig deployment events to Amazon SQS`estensione.

Fase 1: Configurazione AWS AppConfig per accodare i messaggi

Aggiungi una policy Amazon SQS alla tua coda Amazon SQS AWS AppConfig granting `appconfig.amazonaws.com` () send message permissions (). `sqs:SendMessage` Per ulteriori informazioni, consulta Esempi di base delle politiche di Amazon SQS.

Passaggio 2: creare un'associazione di estensioni

Collega l'estensione a una delle tue AWS AppConfig risorse creando un'associazione di estensioni. L'associazione viene creata utilizzando la AWS AppConfig console o l'azione CreateExtensionAssociationAPI. Quando si crea l'associazione, si specifica l'ARN di un' AWS AppConfig applicazione, di un ambiente o di un profilo di configurazione. Se si associa l'estensione a un'applicazione o a un ambiente, viene inviata una notifica per qualsiasi profilo di configurazione contenuto nell'applicazione o nell'ambiente specificato. Quando crei l'associazione, devi inserire un `Here` parametro che contenga l'ARN della coda Amazon SQS che desideri utilizzare.

Dopo aver creato l'associazione, quando viene creata o distribuita una configurazione per la AWS AppConfig risorsa specificata, AWS AppConfig richiama l'estensione e invia notifiche in base ai punti di azione specificati nell'estensione.

> **Note**
>
> Questa estensione viene richiamata dai seguenti punti di azione:
>
> - `ON_DEPLOYMENT_START`
> - `ON_DEPLOYMENT_COMPLETE`
> - `ON_DEPLOYMENT_ROLLED_BACK`
>
> Non puoi personalizzare i punti di azione per questa estensione. Per richiamare diversi punti di azione, puoi creare la tua estensione. Per ulteriori informazioni, consulta Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig.

Utilizzate le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la AWS Systems Manager console o il AWS CLI.

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel riquadro di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Aggiungi alla risorsa.
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di AWS AppConfig risorsa. A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse.
5. Scegli Crea associazione alla risorsa.

Ecco un esempio del messaggio inviato alla coda di Amazon SQS quando viene richiamata l'estensione.

```
{
   "InvocationId":"7itcaxp",
   "Parameters":{
      "queueArn":"arn:aws:sqs:us-east-1:111122223333:MySQSQueue"
   },
   "Application":{
      "Id":"1a2b3c4d",
```

```
      "Name":MyApp
   },
   "Environment":{
      "Id":"1a2b3c4d",
      "Name":MyEnv
   },
   "ConfigurationProfile":{
      "Id":"1a2b3c4d",
      "Name":"MyConfigProfile"
   },
   "Description":null,
   "DeploymentNumber":"3",
   "ConfigurationVersion":"1",
   "Type":"OnDeploymentComplete"
}
```

Utilizzo dell'estensione Atlassian Jira per AWS AppConfig

Grazie all'integrazione con Atlassian Jira, AWS AppConfig puoi creare e aggiornare problemi nella console Atlassian ogni volta che apporti modifiche a un flag di funzionalità nel tuo computer per quanto specificato. Account AWS Regione AWS Ogni numero di Jira include il nome del flag, l'ID dell'applicazione, l'ID del profilo di configurazione e i valori dei flag. Dopo aver aggiornato, salvato e distribuito le modifiche ai flag, Jira aggiorna i problemi esistenti con i dettagli della modifica.

> **Note**
>
> Jira aggiorna i problemi ogni volta che crei o aggiorni un feature flag. Jira aggiorna anche i problemi quando elimini un attributo di flag a livello di figlio da un flag di livello principale. Jira non registra informazioni quando elimini un flag di livello principale.

Per configurare l'integrazione, devi fare quanto segue:

- Configurazione delle autorizzazioni per AWS AppConfig l'integrazione con Jira
- Configurazione dell'applicazione di integrazione AWS AppConfig Jira

Configurazione delle autorizzazioni per AWS AppConfig l'integrazione con Jira

Quando configuri AWS AppConfig l'integrazione con Jira, specifichi le credenziali per un utente. In particolare, inserisci l'ID della chiave di accesso e la chiave segreta dell'utente nell'applicazione AWS AppConfig per Jira. Questo utente concede a Jira il permesso di comunicare con. AWS AppConfig AWS AppConfig utilizza queste credenziali una sola volta per stabilire un'associazione tra AWS AppConfig e Jira. Le credenziali non vengono archiviate. Puoi rimuovere l'associazione disinstallando l'applicazione AWS AppConfig per Jira.

L'account utente richiede una politica di autorizzazione che includa le seguenti azioni:

- `appconfig:CreateExtensionAssociation`
- `appconfig:GetConfigurationProfile`
- `appconfig:ListApplications`
- `appconfig:ListConfigurationProfiles`
- `appconfig:ListExtensionAssociations`
- `sts:GetCallerIdentity`

Completa le seguenti attività per creare una policy di autorizzazione IAM e un utente per AWS AppConfig l'integrazione con Jira:

Attività

- Attività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira
- Attività 2: creare un utente per l'integrazione con Jira AWS AppConfig

Attività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira

Utilizza la seguente procedura per creare una policy di autorizzazione IAM che consenta ad Atlassian Jira di comunicare con. AWS AppConfig Ti consigliamo di creare una nuova policy e di collegarla a un nuovo ruolo IAM. L'aggiunta dell'autorizzazione richiesta a una policy e a un ruolo IAM esistenti è contraria al principio del privilegio minimo e non è consigliata.

Per creare una policy IAM per l'integrazione AWS AppConfig con Jira

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy e quindi Crea policy.

3. Nella pagina Crea policy, scegli la scheda JSON e sostituisci il contenuto predefinito con la seguente policy. Nella seguente politica, sostituisci *Region*, *Account_ID, Application_ID e Configuration_profile_ID* con le informazioni del tuo ambiente di feature flag. AWS AppConfig

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                      "appconfig:CreateExtensionAssociation",
                      "appconfig:ListExtensionAssociations",
                      "appconfig:GetConfigurationProfile"
             ],
            "Resource": [

 "arn:aws:appconfig:Region:account_ID:application/application_ID",

 "arn:aws:appconfig:Region:account_ID:application/application_ID/
configurationprofile/configuration_profile_ID"
             ]
        },
       {
            "Effect": "Allow",
            "Action": [
                      "appconfig:ListApplications"

             ],
            "Resource": [
                      "arn:aws:appconfig:Region:account_ID:*"
             ]
        },
         {
            "Effect": "Allow",
            "Action": [
                      "appconfig:ListConfigurationProfiles"
             ],
            "Resource": [

 "arn:aws:appconfig:Region:account_ID:application/application_ID"
             ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": "sts:GetCallerIdentity",
            "Resource": "*"
        }
    ]
}
```

4. Scegliere Next: Tags (Successivo: Tag).
5. (Facoltativo) Aggiungere una o più coppie tag-valore chiave per organizzare, monitorare o controllare l'accesso per questa policy, quindi scegliere Next: Review (Successivo: Rivedi).
6. Nella pagina Review policy (Rivedi policy), immetti un nome nella casella Name (Nome), ad esempio **`AppConfigJiraPolicy`**, quindi immetti una descrizione facoltativa.
7. Scegli Crea policy.

Attività 2: creare un utente per l'integrazione con Jira AWS AppConfig

Usa la seguente procedura per creare un utente per AWS AppConfig l'integrazione con Atlassian Jira. Dopo aver creato l'utente, puoi copiare l'ID della chiave di accesso e la chiave segreta, che specificherai al termine dell'integrazione.

Per creare un utente AWS AppConfig e un'integrazione con Jira

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Users (Utenti), quindi scegli Add users (Aggiungi utenti).
3. Nel campo Nome utente, inserisci un nome, ad esempio**`AppConfigJiraUser`**.
4. Per Seleziona il tipo di AWS credenziale, scegli Chiave di accesso - Accesso programmatico.
5. Scegli Successivo: Autorizzazioni.
6. Nella pagina Imposta autorizzazioni, scegli Allega direttamente le politiche esistenti. Cerca e seleziona la casella di controllo relativa alla politica in cui hai creatoAttività 1: creare una politica di autorizzazione IAM per l'integrazione AWS AppConfig con Jira, quindi scegli Avanti: tag.
7. Nella pagina Aggiungi tag (opzionale), aggiungi una o più coppie tag-chiave-valore per organizzare, tracciare o controllare l'accesso per questo utente. Seleziona Successivo: Revisione.
8. Nella pagina Revisione, verifica i dettagli dell'utente.

9. Selezionare Create user (Crea utente). Il sistema visualizza l'ID della chiave di accesso e la chiave segreta dell'utente. Scarica il file.csv o copia queste credenziali in una posizione separata. Queste credenziali verranno specificate durante la configurazione dell'integrazione.

Configurazione dell'applicazione di integrazione AWS AppConfig Jira

Usa la seguente procedura per configurare le opzioni richieste nell'applicazione AWS AppConfig per Jira. Dopo aver completato questa procedura, Jira crea un nuovo problema per ogni feature flag presente nel tuo account Account AWS per quanto specificato. Regione AWS Se apporti modifiche a un feature flag in AWS AppConfig, Jira registra i dettagli nei problemi esistenti.

> **Note**
>
> Un AWS AppConfig feature flag può includere più attributi di flag a livello di bambino. Jira crea un problema per ogni feature flag di livello principale. Se modifichi un attributo di flag a livello di figlio, puoi visualizzare i dettagli di tale modifica nel numero di Jira per il flag di livello genitore.

Per configurare l'integrazione

1. Accedi all'Atlassian Marketplace.
2. Digita **`AWS AppConfig`** nel campo di ricerca e premi Invio.
3. Installa l'applicazione sulla tua istanza Jira.
4. Nella console Atlassian, scegli Gestisci app, quindi scegli AWS AppConfig Jira.
5. Scegli Configura.
6. In Dettagli di configurazione, scegli Jira project, quindi scegli il progetto che desideri associare al tuo feature flag. AWS AppConfig
7. Scegli Regione AWS, quindi scegli la regione in cui si trova il tuo AWS AppConfig feature flag.
8. Nel campo ID applicazione, inserisci il nome dell' AWS AppConfig applicazione che contiene il tuo flag di funzionalità.
9. Nel campo ID del profilo di configurazione, inserite il nome del profilo di AWS AppConfig configurazione per il vostro feature flag.
10. Nei campi ID chiave di accesso e Chiave segreta, inserisci le credenziali che hai copiato. Attività 2: creare un utente per l'integrazione con Jira AWS AppConfig Facoltativamente, puoi anche specificare un token di sessione.

11. Seleziona Invia.
12. Nella console Atlassian, scegli Progetti, quindi scegli il progetto selezionato per l'integrazione. AWS AppConfig La pagina Problemi mostra un problema per ogni indicatore di funzionalità nel campo specificato Account AWS e. Regione AWS

Eliminazione dell'applicazione e dei dati AWS AppConfig per Jira

Se non desideri più utilizzare l'integrazione di Jira con i flag di AWS AppConfig funzionalità, puoi eliminare l'applicazione AWS AppConfig per Jira nella console Atlassian. L'eliminazione dell'applicazione di integrazione esegue le seguenti operazioni:

- Elimina l'associazione tra l'istanza di Jira e AWS AppConfig
- Elimina i dettagli dell'istanza Jira da AWS AppConfig

Per eliminare l'applicazione AWS AppConfig for Jira

1. Nella console Atlassian, scegli Gestisci app.
2. Scegli AWS AppConfig per Jira.
3. Scegliere Uninstall (Disinstalla).

Procedura dettagliata: creazione di estensioni personalizzate AWS AppConfig

Per creare un' AWS AppConfig estensione personalizzata, completa le seguenti attività. Ogni attività è descritta più dettagliatamente negli argomenti successivi.

> **Note**
>
> È possibile visualizzare esempi di AWS AppConfig estensioni personalizzate su GitHub:
>
> - Estensione di esempio che impedisce le implementazioni con un calendario di `blocked day` moratoria utilizzando Systems Manager Change Calendar
> - Estensione di esempio che impedisce la divulgazione di segreti nei dati di configurazione utilizzando git-secrets

- Estensione di esempio che impedisce la fuoriuscita di informazioni di identificazione personale (PII) nei dati di configurazione utilizzando Amazon Comprehend

1. Crea una funzione AWS Lambda

   Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Un'eccezione a questa regola è AWS rappresentata dalla creazione di versioni personalizzate delle estensioni di notifica create per aggiungere o rimuovere punti di azione. Per ulteriori dettagli su questa eccezione, consultaCreazione di un'estensione personalizzata AWS AppConfig.

2. Configura le autorizzazioni per la tua estensione personalizzata

   Per configurare le autorizzazioni per l'estensione personalizzata, puoi effettuare una delle seguenti operazioni:

   - Crea un ruolo di servizio AWS Identity and Access Management (IAM) che includa le `InvokeFunction` autorizzazioni.
   - Crea una politica delle risorse utilizzando l'azione AddPermissionAPI Lambda.

   Questa procedura dettagliata descrive come creare il ruolo di servizio IAM.

3. Crea un'estensione

   Puoi creare un'estensione utilizzando la AWS AppConfig console o richiamando l'azione CreateExtensionAPI da AWS CLI AWS Tools for PowerShell, o dall'SDK. La procedura dettagliata utilizza la console.

4. Crea un'associazione di estensioni

   Puoi creare un'associazione di estensioni utilizzando la AWS AppConfig console o richiamando l'azione CreateExtensionAssociationAPI da AWS CLI AWS Tools for PowerShell, o dall'SDK. La procedura dettagliata utilizza la console.

5. Esegui un'azione che richiami l'estensione

   Dopo aver creato l'associazione, AWS AppConfig richiama l'estensione quando si verificano i punti d'azione definiti dall'estensione per quella risorsa. Ad esempio, se si associa un'estensione che contiene un'`PRE_CREATE_HOSTED_CONFIGURATION_VERSION`Nazione, l'estensione viene richiamata ogni volta che si crea una nuova versione di configurazione ospitata.

Gli argomenti di questa sezione descrivono ogni attività coinvolta nella creazione di un' AWS AppConfig estensione personalizzata. Ogni attività viene descritta nel contesto di un caso d'uso in cui un cliente desidera creare un'estensione che esegua automaticamente il backup di una configurazione in un bucket Amazon Simple Storage Service (Amazon S3). L'estensione viene eseguita ogni volta che una configurazione ospitata viene creata (`PRE_CREATE_HOSTED_CONFIGURATION_VERSION`) o distribuita (). `PRE_START_DEPLOYMENT`

Argomenti

Creazione di una funzione Lambda per un'estensione personalizzata AWS AppConfig

Nella maggior parte dei casi d'uso, per creare un'estensione personalizzata, è necessario creare una AWS Lambda funzione per eseguire qualsiasi calcolo ed elaborazione definiti nell'estensione. Questa sezione include il codice di esempio della funzione Lambda per un'estensione personalizzata AWS AppConfig . Questa sezione include anche i dettagli di riferimento della richiesta e della risposta del payload. Per informazioni sulla creazione di una funzione Lambda, consulta Getting started with Lambda nella Developer Guide.AWS Lambda

Codice di esempio

Il seguente codice di esempio per una funzione Lambda, quando richiamato, esegue automaticamente il backup di una AWS AppConfig configurazione in un bucket Amazon S3. Viene eseguito il backup della configurazione ogni volta che viene creata o distribuita una nuova configurazione. L'esempio utilizza i parametri di estensione in modo che il nome del bucket non debba essere codificato nella funzione Lambda. Utilizzando i parametri di estensione, l'utente può collegare l'estensione a più applicazioni ed eseguire il backup delle configurazioni in diversi bucket. L'esempio di codice include commenti per spiegare ulteriormente la funzione.

Esempio di funzione Lambda per un'estensione AWS AppConfig

```
from datetime import datetime
```

```
import base64
import json

import boto3


def lambda_handler(event, context):
    print(event)

    # Extensions that use the PRE_CREATE_HOSTED_CONFIGURATION_VERSION and
 PRE_START_DEPLOYMENT
    # action points receive the contents of AWS AppConfig configurations in Lambda
 event parameters.
    # Configuration contents are received as a base64-encoded string, which the lambda
 needs to decode
    # in order to get the configuration data as bytes. For other action points, the
 content
    # of the configuration isn't present, so the code below will fail.
    config_data_bytes = base64.b64decode(event["Content"])

    # You can specify parameters for extensions. The CreateExtension API action lets
 you define
    # which parameters an extension supports. You supply the values for those
 parameters when you
    # create an extension association by calling the CreateExtensionAssociation API
 action.
    # The following code uses a parameter called S3_BUCKET to obtain the value
 specified in the
    # extension association. You can specify this parameter when you create the
 extension
    # later in this walkthrough.
    extension_association_params = event.get('Parameters', {})
    bucket_name = extension_association_params['S3_BUCKET']
    write_backup_to_s3(bucket_name, config_data_bytes)

    # The PRE_CREATE_HOSTED_CONFIGURATION_VERSION and PRE_START_DEPLOYMENT action
 points can
    # modify the contents of a configuration. The following code makes a minor change
    # for the purposes of a demonstration.
    old_config_data_string = config_data_bytes.decode('utf-8')
    new_config_data_string = old_config_data_string.replace('hello', 'hello!')
    new_config_data_bytes = new_config_data_string.encode('utf-8')

    # The lambda initially received the configuration data as a base64-encoded string
```

```
    # and must return it in the same format.
    new_config_data_base64string =
 base64.b64encode(new_config_data_bytes).decode('ascii')

    return {
        'statusCode': 200,
        # If you want to modify the contents of the configuration, you must include the
 new contents in the
        # Lambda response. If you don't want to modify the contents, you can omit the
 'Content' field shown here.
        'Content': new_config_data_base64string
    }


def write_backup_to_s3(bucket_name, config_data_bytes):
    s3 = boto3.resource('s3')
    new_object = s3.Object(bucket_name,
 f"config_backup_{datetime.now().isoformat()}.txt")
    new_object.put(Body=config_data_bytes)
```

Se desideri utilizzare questo esempio durante questa procedura dettagliata, salvalo con il nome **`MyS3ConfigurationBackUpExtension`** e copia l'Amazon Resource Name (ARN) per la funzione. L'ARN viene specificato quando si crea il ruolo di assunzione AWS Identity and Access Management (IAM) nella sezione successiva. L'ARN e il nome vengono specificati al momento della creazione dell'estensione.

Riferimento del payload

Questa sezione include i dettagli di riferimento della richiesta di payload e della risposta per l'utilizzo delle estensioni personalizzate AWS AppConfig .

Struttura della richiesta

PreCreateHostedConfigurationVersion

```
{
    'InvocationId': 'vlns753', // id for specific invocation
    'Parameters': {
        'ParameterOne': 'ValueOne',
        'ParameterTwo': 'ValueTwo'
    },
    'ContentType': 'text/plain',
```

```
    'ContentVersion': '2',
    'Content': 'SGVsbG8gZWFydGgh', // Base64 encoded content
    'Application': {
        'Id': 'abcd123',
        'Name': 'ApplicationName'
    },
    'ConfigurationProfile': {
        'Id': 'ijkl789',
        'Name': 'ConfigurationName'
    },
    'Description': '',
    'Type': 'PreCreateHostedConfigurationVersion',
    'PreviousContent': {
        'ContentType': 'text/plain',
        'ContentVersion': '1',
        'Content': 'SGVsbG8gd29ybGQh'
    }
}
```

PreStartDeployment

```
{
    'InvocationId': '765ahdm',
    'Parameters': {
        'ParameterOne': 'ValueOne',
        'ParameterTwo': 'ValueTwo'
    },
    'ContentType': 'text/plain',
    'ContentVersion': '2',
    'Content': 'SGVsbG8gZWFydGgh',
    'Application': {
        'Id': 'abcd123',
        'Name': 'ApplicationName'
    },
    'Environment': {
        'Id': 'ibpnqlq',
        'Name': 'EnvironmentName'
    },
    'ConfigurationProfile': {
        'Id': 'ijkl789',
        'Name': 'ConfigurationName'
    },
    'DeploymentNumber': 2,
```

```
    'Description': 'Deployment description',
    'Type': 'PreStartDeployment'
}
```

Eventi asincroni

OnStartDeployment, OnDeploymentStep, OnDeployment

```
{
    'InvocationId': 'o2xbtm7',
    'Parameters': {
        'ParameterOne': 'ValueOne',
        'ParameterTwo': 'ValueTwo'
    },
    'Type': 'OnDeploymentStart',
    'Application': {
        'Id': 'abcd123'
    },
    'Environment': {
        'Id': 'efgh456'
    },
    'ConfigurationProfile': {
        'Id': 'ijkl789',
        'Name': 'ConfigurationName'
    },
    'DeploymentNumber': 2,
    'Description': 'Deployment description',
    'ConfigurationVersion': '2'
}
```

Struttura della risposta

Gli esempi seguenti mostrano cosa restituisce la funzione Lambda in risposta alla richiesta di un'estensione personalizzata. AWS AppConfig

Eventi sincroni: risposta riuscita

Se vuoi trasformare il contenuto, usa quanto segue:

```
"Content": "SomeBase64EncodedByteArray"
```

Se non vuoi trasformare il contenuto, non restituisci nulla.

Eventi asincroni: risposta riuscita

Non restituisci nulla.

Tutti gli eventi di errore

```
{
        "Error": "BadRequestError",
        "Message": "There was malformed stuff in here",
        "Details": [{
            "Type": "Malformed",
            "Name": "S3 pointer",
            "Reason": "S3 bucket did not exist"
        }]
    }
```

Configurazione delle autorizzazioni per un'estensione personalizzata AWS AppConfig

Utilizza la seguente procedura per creare e configurare un ruolo di servizio AWS Identity and Access Management (IAM) (o assumere un ruolo). AWS AppConfig utilizza questo ruolo per richiamare la funzione Lambda.

Per creare un ruolo di servizio IAM e consentire di AWS AppConfig assumerlo

1. Apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli e quindi Crea ruolo.
3. In Seleziona il tipo di entità affidabile, scegli Politica di fiducia personalizzata.
4. Incolla la seguente politica JSON nel campo Politica di fiducia personalizzata.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "appconfig.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

Seleziona Avanti.

5. Nella pagina Aggiungi autorizzazioni, scegli Crea politica. La pagina Create policy (Crea policy) viene aperta in una nuova scheda.
6. Scegli la scheda JSON, quindi incolla la seguente politica di autorizzazione nell'editor. L'`lambda:InvokeFunction`azione viene utilizzata per i punti `PRE_*` d'azione. L'`lambda:InvokeAsync`azione viene utilizzata per i punti `ON_*` azione. Sostituisci *`il tuo Lambda ARN con`* l'Amazon Resource Name (ARN) della tua Lambda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction",
        "lambda:InvokeAsync"
      ],
      "Resource": "Your Lambda ARN"
    }
  ]
}
```

7. Scegliere Successivo: Tag.
8. Nella pagina Aggiungi tag (opzionale), aggiungi una o più coppie chiave-valore, quindi scegli Avanti: revisione.
9. Nella pagina Revisione della politica, inserisci un nome e una descrizione, quindi scegli Crea politica.
10. Nella scheda del browser relativa alla politica di attendibilità personalizzata, scegli l'icona Aggiorna, quindi cerca la politica di autorizzazione appena creata.
11. Seleziona la casella di controllo relativa alla politica di autorizzazione, quindi scegli Avanti.
12. Nella pagina Nome, rivedi e crea, inserisci un nome nella casella Nome ruolo, quindi inserisci una descrizione.
13. Scegliere Create role (Crea ruolo). Il sistema visualizza di nuovo la pagina Roles (Ruoli). Scegli Visualizza ruolo nel banner.

14. Copia l'ARN. Specificate questo ARN quando create l'estensione.

Creazione di un'estensione personalizzata AWS AppConfig

Un'estensione definisce una o più azioni che esegue durante un AWS AppConfig flusso di lavoro. Ad esempio, l'`AWS AppConfig deployment events to Amazon SNS`estensione AWS creata include un'azione per inviare una notifica a un argomento di Amazon SNS. Ogni azione viene richiamata quando interagisci con AWS AppConfig o quando AWS AppConfig esegui un processo per tuo conto. Questi sono chiamati punti d'azione. AWS AppConfig le estensioni supportano i seguenti punti di azione:

- `PRE_CREATE_HOSTED_CONFIGURATION_VERSION`
- `PRE_START_DEPLOYMENT`
- `ON_DEPLOYMENT_START`
- `ON_DEPLOYMENT_STEP`
- `ON_DEPLOYMENT_BAKING`
- `ON_DEPLOYMENT_COMPLETE`
- `ON_DEPLOYMENT_ROLLED_BACK`

Le azioni di estensione configurate sui punti di `PRE_*` azione vengono applicate dopo la convalida della richiesta, ma AWS AppConfig prima dell'esecuzione dell'attività corrispondente al nome del punto di azione. Queste chiamate di azione vengono elaborate contemporaneamente a una richiesta. Se viene effettuata più di una richiesta, le chiamate alle azioni vengono eseguite in sequenza. Si noti inoltre che i punti di `PRE_*` azione ricevono e possono modificare il contenuto di una configurazione. `PRE_*`i punti di azione possono anche rispondere a un errore e impedire che si verifichi un'azione.

Un'estensione può anche essere eseguita in parallelo a un AWS AppConfig flusso di lavoro utilizzando un punto di `ON_*` azione. `ON_*`i punti di azione vengono richiamati in modo asincrono. `ON_*`i punti di azione non ricevono il contenuto di una configurazione. Se un'estensione riscontra un errore durante un punto di `ON_*` azione, il servizio ignora l'errore e continua il flusso di lavoro.

L'estensione di esempio seguente definisce un'azione che richiama il punto di `PRE_CREATE_HOSTED_CONFIGURATION_VERSION` azione. Nel `Uri` campo, l'azione specifica l'Amazon Resource Name (ARN) della funzione `MyS3ConfigurationBackUpExtension` Lambda creata in precedenza in questa procedura dettagliata. L'azione specifica anche l'ARN di assunzione

del ruolo AWS Identity and Access Management (IAM) creato in precedenza in questa procedura dettagliata.

Estensione di esempio AWS AppConfig

```
{
    "Name": "MySampleExtension",
    "Description": "A sample extension that backs up configurations to an S3 bucket.",
    "Actions": {
        "PRE_CREATE_HOSTED_CONFIGURATION_VERSION": [
            {
                "Name": "PreCreateHostedConfigVersionActionForS3Backup",
                "Uri": "arn:aws:lambda:aws-
region:111122223333:function:MyS3ConfigurationBackUpExtension",
                "RoleArn": "arn:aws:iam::111122223333:role/ExtensionsTestRole"
            }
        ]
    },
    "Parameters" : {
        "S3_BUCKET": {
            "Required": false
        }
    }
}
```

> **Note**
>
> Per visualizzare la sintassi della richiesta e le descrizioni dei campi durante la creazione di un'estensione, consulta l'CreateExtensionargomento nell'AWS AppConfig API Reference.

Per creare un'estensione (console)

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel riquadro di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli Crea estensione.
4. Per Nome estensione, inserisci un nome univoco. Ai fini di questa procedura dettagliata, immettere. **`MyS3ConfigurationBackUpExtension`** Facoltativamente, inserire una descrizione.

5. Nella sezione Azioni, scegli Aggiungi nuova azione.
6. Per Nome azione, inserisci un nome univoco. Ai fini di questa procedura dettagliata, immettere. **`PreCreateHostedConfigVersionActionForS3Backup`** Questo nome descrive il punto d'azione utilizzato dall'azione e lo scopo dell'estensione.
7. Nell'elenco dei punti d'azione, scegliete PRE_CREATE_HOSTED_CONFIGURATION_VERSION.
8. Per Uri, scegli la funzione Lambda, quindi scegli la funzione nell'elenco delle funzioni Lambda. Se non vedi la tua funzione, verifica di trovarti nella stessa posizione in Regione AWS cui l'hai creata.
9. Per IAM Role, scegli il ruolo che hai creato in precedenza in questa procedura dettagliata.
10. Nella sezione Parametri di estensione (opzionale), scegli Aggiungi nuovo parametro.
11. Per Nome del parametro, inserisci un nome. Ai fini di questa procedura dettagliata, immettere. **`S3_BUCKET`**
12. Ripetete i passaggi da 5 a 11 per creare una seconda azione per il punto d'azione. `PRE_START_DEPLOYMENT`
13. Scegliete Crea estensione.

Personalizzazione delle estensioni di AWS notifica create

Non è necessario creare una Lambda o un'estensione per utilizzare le estensioni di notifica AWS create. Puoi semplicemente creare un'associazione di estensioni e quindi eseguire un'operazione che richiama uno dei punti di azione supportati. Per impostazione predefinita, le AWS estensioni di notifica create supportano i seguenti punti di azione:

- `ON_DEPLOYMENT_START`
- `ON_DEPLOYMENT_COMPLETE`
- `ON_DEPLOYMENT_ROLLED_BACK`

Se crei versioni personalizzate dell'`AWS AppConfig deployment events to Amazon SNS`estensione e delle `AWS AppConfig deployment events to Amazon SQS` estensioni, puoi specificare i punti di azione per i quali desideri ricevere notifiche.

> **Note**
>
> L'`AWS AppConfig deployment events to EventBridge`estensione non supporta i punti di `PRE_*` azione. Puoi creare una versione personalizzata se desideri rimuovere alcuni dei punti di azione predefiniti assegnati alla AWS versione creata.

Non è necessario creare una funzione Lambda se si creano versioni personalizzate delle estensioni di notifica AWS create. Devi solo specificare un Amazon Resource Name (ARN) nel `Uri` campo per la nuova versione dell'estensione.

- Per un'estensione di EventBridge notifica personalizzata, inserisci l'ARN degli eventi EventBridge predefiniti nel `Uri` campo.
- Per un'estensione di notifica Amazon SNS personalizzata, inserisci l'ARN di un argomento Amazon SNS nel campo. `Uri`
- Per un'estensione di notifica Amazon SQS personalizzata, inserisci l'ARN di una coda di messaggi Amazon SQS nel campo. `Uri`

Creazione di un'associazione di estensioni per un'estensione personalizzata AWS AppConfig

Per creare un'estensione o configurare un' AWS estensione creata, si definiscono i punti di azione che richiamano un'estensione quando viene utilizzata una AWS AppConfig risorsa specifica. Ad esempio, puoi scegliere di eseguire l'`AWS AppConfig deployment events to Amazon SNS`estensione e ricevere notifiche su un argomento di Amazon SNS ogni volta che viene avviata una distribuzione di configurazione per un'applicazione specifica. La definizione dei punti di azione che richiamano un'estensione per una AWS AppConfig risorsa specifica viene chiamata associazione di estensioni. Un'associazione di estensione è una relazione specifica tra un'estensione e una AWS AppConfig risorsa, ad esempio un'applicazione o un profilo di configurazione.

Una singola AWS AppConfig applicazione può includere più ambienti e profili di configurazione. Se si associa un'estensione a un'applicazione o a un ambiente, AWS AppConfig richiama l'estensione per tutti i flussi di lavoro relativi all'applicazione o alle risorse dell'ambiente, se applicabile.

Ad esempio, supponiamo di avere un' AWS AppConfig applicazione chiamata MobileApps che include un profilo di configurazione chiamato. AccessList Supponiamo che l' MobileApps applicazione includa ambienti beta, di integrazione e di produzione. Crei un'associazione di estensione per l'

AWS estensione di notifica Amazon SNS creata e associ l'estensione all' MobileApps applicazione. L'estensione di notifica di Amazon SNS viene richiamata ogni volta che la configurazione viene distribuita per l'applicazione in uno dei tre ambienti.

Utilizza le seguenti procedure per creare un'associazione di AWS AppConfig estensioni utilizzando la console. AWS AppConfig

Per creare un'associazione di estensioni (console)

1. Apri la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/appconfig/.
2. Nel riquadro di navigazione, scegli AWS AppConfig.
3. Nella scheda Estensioni, scegli un pulsante di opzione per un'estensione, quindi scegli Aggiungi alla risorsa. Ai fini di questa procedura dettagliata, scegli MyS3. ConfigurationBackUpExtension
4. Nella sezione Dettagli della risorsa di estensione, per Tipo di risorsa, scegli un tipo di risorsa. AWS AppConfig A seconda della risorsa scelta, AWS AppConfig ti chiede di scegliere altre risorse. Ai fini di questa procedura dettagliata, scegli Applicazione.
5. Scegli un'applicazione nell'elenco.
6. Nella sezione Parametri, verifica che S3_BUCKET sia elencato nel campo Chiave. Nel campo Valore, incolla l'ARN delle estensioni Lambda. Ad esempio: `arn:aws:lambda:`*`aws-region`*`:111122223333:function:MyS3ConfigurationBackUpExtension`.
7. Scegli Crea associazione alla risorsa.

Esecuzione di un'azione che richiama un'estensione personalizzata AWS AppConfig

Dopo aver creato l'associazione, è possibile richiamare l'`MyS3ConfigurationBackUpExtension`estensione creando un nuovo profilo di configurazione che lo specifichi`hosted`. `SourceUri` Come parte del flusso di lavoro per creare la nuova configurazione, AWS AppConfig incontra il punto di azione. `PRE_CREATE_HOSTED_CONFIGURATION_VERSION` L'incontro di questo punto di azione richiama l'`MyS3ConfigurationBackUpExtension`estensione, che esegue automaticamente il backup della configurazione appena creata nel bucket S3 specificato nella sezione dell'associazione di estensione. `Parameter`

AWS AppConfig integrazione delle estensioni con Atlassian Jira

AWS AppConfig si integra con Atlassian Jira. L'integrazione consente di AWS AppConfig creare e aggiornare problemi nella console Atlassian ogni volta che si apportano modifiche a un flag di funzionalità nel campo specificato. Account AWS Regione AWS Ogni numero di Jira include il nome del flag, l'ID dell'applicazione, l'ID del profilo di configurazione e i valori dei flag. Dopo aver aggiornato, salvato e distribuito le modifiche ai flag, Jira aggiorna i problemi esistenti con i dettagli della modifica. Per ulteriori informazioni, consulta Utilizzo dell'estensione Atlassian Jira per AWS AppConfig.

Codici di esempio AWS AppConfig

Questa sezione include esempi di codice per eseguire azioni comuni AWS AppConfig a livello di codice. Ti consigliamo di utilizzare questi esempi con Java, Python e JavaScriptSDK per eseguire le azioni in un ambiente di test. Questa sezione include un esempio di codice per ripulire l'ambiente di test al termine.

Argomenti

Creazione o aggiornamento di una configurazione in formato libero archiviata nell'archivio di configurazione ospitato

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano le seguenti API:

- CreateApplication
- CreateConfigurationProfile
- CreateHostedConfigurationVersion

Java

```
public CreateHostedConfigurationVersionResponse createHostedConfigVersion() {
        AppConfigClient appconfig = AppConfigClient.create();

        // Create an application
```

```
        CreateApplicationResponse app = appconfig.createApplication(req ->
 req.name("MyDemoApp"));

        // Create a hosted, freeform configuration profile
        CreateConfigurationProfileResponse configProfile =
 appconfig.createConfigurationProfile(req -> req
            .applicationId(app.id())
            .name("MyConfigProfile")
            .locationUri("hosted")
            .type("AWS.Freeform"));

        // Create a hosted configuration version
        CreateHostedConfigurationVersionResponse hcv =
 appconfig.createHostedConfigurationVersion(req -> req
            .applicationId(app.id())
            .configurationProfileId(configProfile.id())
            .contentType("text/plain; charset=utf-8")
            .content(SdkBytes.fromUtf8String("my config data")));

        return hcv;
    }
```

Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')

# create a hosted, freeform configuration profile
config_profile = appconfig.create_configuration_profile(
    ApplicationId=application['Id'],
    Name='MyConfigProfile',
    LocationUri='hosted',
    Type='AWS.Freeform')

# create a hosted configuration version
hcv = appconfig.create_hosted_configuration_version(
    ApplicationId=application['Id'],
    ConfigurationProfileId=config_profile['Id'],
    Content=b'my config data',
```

```
    ContentType='text/plain')
```

JavaScript

```
import {
  AppConfigClient,
  CreateApplicationCommand,
  CreateConfigurationProfileCommand,
  CreateHostedConfigurationVersionCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
);

// create a hosted, freeform configuration profile
const profile = await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "hosted",
    Type: "AWS.Freeform",
  })
);

// create a hosted configuration version
await appconfig.send(
  new CreateHostedConfigurationVersionCommand({
    ApplicationId: application.Id,
    ConfigurationProfileId: profile.Id,
    ContentType: "text/plain",
    Content: "my config data",
  })
);
```

Creazione di un profilo di configurazione per un segreto archiviato in Secrets Manager

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano le seguenti API:

- CreateApplication
- CreateConfigurationProfile

Java

```
private void createSecretsManagerConfigProfile() {
        AppConfigClient appconfig = AppConfigClient.create();

        // Create an application
        CreateApplicationResponse app = appconfig.createApplication(req ->
 req.name("MyDemoApp"));

        // Create a configuration profile for Secrets Manager Secret
        CreateConfigurationProfileResponse configProfile =
 appconfig.createConfigurationProfile(req -> req
            .applicationId(app.id())
            .name("MyConfigProfile")
            .locationUri("secretsmanager://MySecret")
            .retrievalRoleArn("arn:aws:iam::000000000000:role/
RoleTrustedByAppConfigThatCanRetrieveSecret")
            .type("AWS.Freeform"));
    }
```

Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')

# create a configuration profile for Secrets Manager Secret
config_profile = appconfig.create_configuration_profile(
```

```
    ApplicationId=application['Id'],
    Name='MyConfigProfile',
    LocationUri='secretsmanager://MySecret',
    RetrievalRoleArn='arn:aws:iam::000000000000:role/
RoleTrustedByAppConfigThatCanRetrieveSecret',
    Type='AWS.Freeform')
```

JavaScript

```
import {
  AppConfigClient,
  CreateConfigurationProfileCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
);

// create a configuration profile for Secrets Manager Secret
await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "secretsmanager://MySecret",
    RetrievalRoleArn: "arn:aws:iam::000000000000:role/
RoleTrustedByAppConfigThatCanRetrieveSecret",
    Type: "AWS.Freeform",
  })
);
```

Implementazione di un profilo di configurazione

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano le seguenti API:

- CreateApplication
- CreateConfigurationProfile
- CreateHostedConfigurationVersion

- CreateEnvironment
- StartDeployment
- GetDeployment

Java

```
private void createDeployment() throws InterruptedException {
        AppConfigClient appconfig = AppConfigClient.create();

        // Create an application
        CreateApplicationResponse app = appconfig.createApplication(req ->
 req.name("MyDemoApp"));

        // Create a hosted, freeform configuration profile
        CreateConfigurationProfileResponse configProfile =
 appconfig.createConfigurationProfile(req -> req
            .applicationId(app.id())
            .name("MyConfigProfile")
            .locationUri("hosted")
            .type("AWS.Freeform"));

        // Create a hosted configuration version
        CreateHostedConfigurationVersionResponse hcv =
 appconfig.createHostedConfigurationVersion(req -> req
            .applicationId(app.id())
            .configurationProfileId(configProfile.id())
            .contentType("text/plain; charset=utf-8")
            .content(SdkBytes.fromUtf8String("my config data")));


        // Create an environment
        CreateEnvironmentResponse env = appconfig.createEnvironment(req -> req
            .applicationId(app.id())
            .name("Beta")
            // If you have CloudWatch alarms that monitor the health of your
 service, you can add them here and they
            // will trigger a rollback if they fire during an appconfig deployment
            //.monitors(Monitor.builder().alarmArn("arn:aws:cloudwatch:us-
east-1:520900602629:alarm:MyAlarm")
            //
  .alarmRoleArn("arn:aws:iam::520900602629:role/MyAppConfigAlarmRole").build())
        );
```

```
        // Start a deployment
        StartDeploymentResponse deploymentResponse = appconfig.startDeployment(req -
> req
            .applicationId(app.id())
            .configurationProfileId(configProfile.id())
            .environmentId(env.id())
            .configurationVersion(hcv.versionNumber().toString())
            .deploymentStrategyId("AppConfig.Linear50PercentEvery30Seconds")
        );

        // Wait for deployment to complete
        List<DeploymentState> nonFinalDeploymentStates = Arrays.asList(
            DeploymentState.DEPLOYING,
            DeploymentState.BAKING,
            DeploymentState.ROLLING_BACK,
            DeploymentState.VALIDATING);
        GetDeploymentRequest getDeploymentRequest =
 GetDeploymentRequest.builder().applicationId(app.id())

 .environmentId(env.id())

 .deploymentNumber(deploymentResponse.deploymentNumber()).build();
        GetDeploymentResponse deployment =
 appconfig.getDeployment(getDeploymentRequest);
        while (nonFinalDeploymentStates.contains(deployment.state())) {
            System.out.println("Waiting for deployment to complete: " + deployment);
            Thread.sleep(1000L);
            deployment = appconfig.getDeployment(getDeploymentRequest);
        }

        System.out.println("Deployment complete: " + deployment);
    }
```

Python

```
import boto3

appconfig = boto3.client('appconfig')

# create an application
application = appconfig.create_application(Name='MyDemoApp')
```

```
# create an environment
environment = appconfig.create_environment(
    ApplicationId=application['Id'],
    Name='MyEnvironment')

# create a configuration profile
config_profile = appconfig.create_configuration_profile(
    ApplicationId=application['Id'],
    Name='MyConfigProfile',
    LocationUri='hosted',
    Type='AWS.Freeform')

# create a hosted configuration version
hcv = appconfig.create_hosted_configuration_version(
    ApplicationId=application['Id'],
    ConfigurationProfileId=config_profile['Id'],
    Content=b'my config data',
    ContentType='text/plain')

# start a deployment
deployment = appconfig.start_deployment(
    ApplicationId=application['Id'],
    EnvironmentId=environment['Id'],
    ConfigurationProfileId=config_profile['Id'],
    ConfigurationVersion=str(hcv['VersionNumber']),
    DeploymentStrategyId='AppConfig.Linear20PercentEvery6Minutes')
```

JavaScript

```
import {
  AppConfigClient,
  CreateApplicationCommand,
  CreateEnvironmentCommand,
  CreateConfigurationProfileCommand,
  CreateHostedConfigurationVersionCommand,
  StartDeploymentCommand,
} from "@aws-sdk/client-appconfig";

const appconfig = new AppConfigClient();

// create an application
const application = await appconfig.send(
  new CreateApplicationCommand({ Name: "MyDemoApp" })
```

```
);

// create an environment
const environment = await appconfig.send(
  new CreateEnvironmentCommand({
    ApplicationId: application.Id,
    Name: "MyEnvironment",
  })
);

// create a configuration profile
const config_profile = await appconfig.send(
  new CreateConfigurationProfileCommand({
    ApplicationId: application.Id,
    Name: "MyConfigProfile",
    LocationUri: "hosted",
    Type: "AWS.Freeform",
  })
);

// create a hosted configuration version
const hcv = await appconfig.send(
  new CreateHostedConfigurationVersionCommand({
    ApplicationId: application.Id,
    ConfigurationProfileId: config_profile.Id,
    Content: "my config data",
    ContentType: "text/plain",
  })
);

// start a deployment
await appconfig.send(
  new StartDeploymentCommand({
    ApplicationId: application.Id,
    EnvironmentId: environment.Id,
    ConfigurationProfileId: config_profile.Id,
    ConfigurationVersion: hcv.VersionNumber.toString(),
    DeploymentStrategyId: "AppConfig.Linear20PercentEvery6Minutes",
  })
);
```

Utilizzo di AWS AppConfig Agent per leggere un profilo di configurazione in formato libero

Ciascuno degli esempi seguenti include commenti sulle azioni eseguite dal codice.

Java

```
public void retrieveConfigFromAgent() throws Exception {
        /*
        In this sample, we will retrieve configuration data from the AWS AppConfig
 Agent.
        The agent is a sidecar process that handles retrieving configuration data
 from AppConfig
        for you in a way that implements best practices like configuration caching.

        For more information about the agent, see Simplified retrieval methods
        */

        // The agent runs a local HTTP server that serves configuration data
        // Make a GET request to the agent's local server to retrieve the
 configuration data
        URL url = new URL("http://localhost:2772/applications/MyDemoApp/
environments/Beta/configurations/MyConfigProfile");
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("GET");
        StringBuilder content;
        try (BufferedReader in = new BufferedReader(new
 InputStreamReader(con.getInputStream()))) {
            content = new StringBuilder();
            int ch;
            while ((ch = in.read()) != -1) {
                content.append((char) ch);
            }
        }
        con.disconnect();
        System.out.println("Configuration from agent via HTTP: " + content);
    }
```

Python

```
# in this sample, we will retrieve configuration data from the AWS AppConfig Agent.
```

```
# the agent is a sidecar process that handles retrieving configuration data from AWS
 AppConfig
# for you in a way that implements best practices like configuration caching.
#
# for more information about the agent, see
# Simplified retrieval methods
#

import requests

application_name = 'MyDemoApp'
environment_name = 'MyEnvironment'
config_profile_name = 'MyConfigProfile'

# the agent runs a local HTTP server that serves configuration data
# make a GET request to the agent's local server to retrieve the configuration data
response = requests.get(f"http://localhost:2772/applications/{application_name}/
environments/{environment_name}/configurations/{config_profile_name}")
config = response.content
```

JavaScript

```
// in this sample, we will retrieve configuration data from the AWS AppConfig Agent.
// the agent is a sidecar process that handles retrieving configuration data from
 AppConfig
// for you in a way that implements best practices like configuration caching.

// for more information about the agent, see
// Simplified retrieval methods

const application_name = "MyDemoApp";
const environment_name = "MyEnvironment";
const config_profile_name = "MyConfigProfile";

// the agent runs a local HTTP server that serves configuration data
// make a GET request to the agent's local server to retrieve the configuration data
const url = `http://localhost:2772/applications/${application_name}/environments/
${environment_name}/configurations/${config_profile_name}`;
const response = await fetch(url);
const config = await response.text(); // (use `await response.json()` if your config
 is json)
```

Utilizzo di AWS AppConfig Agent per leggere un indicatore di funzionalità specifico

Ciascuno dei seguenti esempi include commenti sulle azioni eseguite dal codice.

Java

```
public void retrieveSingleFlagFromAgent() throws Exception {
        /*
          You can retrieve a single flag's data from the agent by providing the
 "flag" query string parameter.
          Note: the configuration's type must be AWS.AppConfig.FeatureFlags
        */

        URL url = new URL("http://localhost:2772/applications/MyDemoApp/
environments/Beta/configurations/MyFlagsProfile?flag=myFlagKey");
        HttpURLConnection con = (HttpURLConnection) url.openConnection();
        con.setRequestMethod("GET");
        StringBuilder content;
        try (BufferedReader in = new BufferedReader(new
 InputStreamReader(con.getInputStream()))) {
            content = new StringBuilder();
            int ch;
            while ((ch = in.read()) != -1) {
                content.append((char) ch);
            }
        }
        con.disconnect();
        System.out.println("MyFlagName from agent: " + content);
    }
```

Python

```
import requests

application_name = 'MyDemoApp'
environment_name = 'MyEnvironment'
config_profile_name = 'MyConfigProfile'
flag_key = 'MyFlag'

# retrieve a single flag's data by providing the "flag" query string parameter
# note: the configuration's type must be AWS.AppConfig.FeatureFlags
```

```
response = requests.get(f"http://localhost:2772/applications/{application_name}/
environments/{environment_name}/configurations/{config_profile_name}?
flag={flag_key}")
config = response.content
```

JavaScript

```
const application_name = "MyDemoApp";
const environment_name = "MyEnvironment";
const config_profile_name = "MyConfigProfile";
const flag_name = "MyFlag";

// retrieve a single flag's data by providing the "flag" query string parameter
// note: the configuration's type must be AWS.AppConfig.FeatureFlags
const url = `http://localhost:2772/applications/${application_name}/environments/
${environment_name}/configurations/${config_profile_name}?flag=${flag_name}`;
const response = await fetch(url);
const flag = await response.json(); // { "enabled": true/false }
```

Utilizzo dell'azione GetLatestConfig API per leggere un profilo di configurazione in formato libero

Ciascuno degli esempi seguenti include commenti sulle azioni eseguite dal codice. Gli esempi di questa sezione richiamano le seguenti API:

- GetLatestConfiguration
- StartConfigurationSession

Java

```
public void retrieveConfigFromApi() {
        /*
        The example below uses two AppConfigData APIs: StartConfigurationSession and
 GetLatestConfiguration.
        For more information on these APIs, see AWS AppConfig Data         */
        AppConfigDataClient appConfigData = AppConfigDataClient.create();

        /*
```

```
        Start a new configuration session using the StartConfigurationSession API.
This operation does not return configuration data.
        Rather, it returns an initial configuration token that should be passed to
GetLatestConfiguration.
        IMPORTANT: This operation should only be performed once (per configuration),
prior to the first GetLatestConfiguration
        call you preform. Each GetLatestConfiguration will return a new
configuration token that you should then use in the
        next GetLatestConfiguration call.
        */
        StartConfigurationSessionResponse session =
            appConfigData.startConfigurationSession(req -> req
                .applicationIdentifier("MyDemoApp")
                .configurationProfileIdentifier("MyConfigProfile")
                .environmentIdentifier("Beta"));

        /*
        Retrieve configuration data using the GetLatestConfiguration API. The first
time you call this API your configuration
        data will be returned. You should cache that data (and the configuration
token) and update that cache asynchronously
        by regularly polling the GetLatestConfiguration API in a background thread.
If you already have the latest configuration
        data, subsequent GetLatestConfiguration calls will return an empty response.
If you then deploy updated configuration
        data the next time you call GetLatestConfiguration it will return that
updated data.

        You can also avoid all the complexity around writing this code yourself by
leveraging our agent instead.
        For more information about the agent, see Simplified retrieval methods
        */

        // The first getLatestConfiguration call uses the token from
StartConfigurationSession
        String configurationToken = session.initialConfigurationToken();
        GetLatestConfigurationResponse configuration =

appConfigData.getLatestConfiguration(GetLatestConfigurationRequest.builder().configurationT

        System.out.println("Configuration retrieved via API: " +
configuration.configuration().asUtf8String());
```

```
        // You'll want to hold on to the token in the getLatestConfiguration
 response because you'll need to use it
        // the next time you call
        configurationToken = configuration.nextPollConfigurationToken();
        configuration =

 appConfigData.getLatestConfiguration(GetLatestConfigurationRequest.builder().configurationT

        // Try creating a new deployment at this point to see how the output below
 changes.
        if (configuration.configuration().asByteArray().length != 0) {
            System.out.println("Configuration contents have changed
 since the last GetLatestConfiguration call, new contents = " +
 configuration.configuration().asUtf8String());
        } else {
            System.out.println("GetLatestConfiguration returned an empty response
 because we already have the latest configuration");
        }
    }
```

Python

```
# the example below uses two AppConfigData APIs: StartConfigurationSession and
 GetLatestConfiguration.
#
# for more information on these APIs, see
# AWS AppConfig Data
#

import boto3

application_name = 'MyDemoApp'
environment_name = 'MyEnvironment'
config_profile_name = 'MyConfigProfile'

appconfigdata = boto3.client('appconfigdata')

# start a new configuration session.
# this operation does not return configuration data.
# rather, it returns an initial configuration token that should be passed to
 GetLatestConfiguration.
#
# note: this operation should only be performed once (per configuration).
```

```
#   all subsequent calls to AppConfigData should be via GetLatestConfiguration.
scs = appconfigdata.start_configuration_session(
    ApplicationIdentifier=application_name,
    EnvironmentIdentifier=environment_name,
    ConfigurationProfileIdentifier=config_profile_name)
initial_token = scs['InitialConfigurationToken']

# retrieve configuration data from the session.
# this operation returns your configuration data.
# each invocation of this operation returns a unique token that should be passed to
 the subsequent invocation.
#
# note: this operation does not always return configuration data after the first
 invocation.
#   data is only returned if the configuration has changed within AWS AppConfig
 (i.e. a deployment occurred).
#   therefore, you should cache the data returned by this call so that you can use
 it later.
glc = appconfigdata.get_latest_configuration(ConfigurationToken=initial_token)
config = glc['Configuration'].read()
```

JavaScript

```
// the example below uses two AppConfigData APIs: StartConfigurationSession and
 GetLatestConfiguration.

// for more information on these APIs, see
// AWS AppConfig Data

import {
  AppConfigDataClient,
  GetLatestConfigurationCommand,
  StartConfigurationSessionCommand,
} from "@aws-sdk/client-appconfigdata";

const appconfigdata = new AppConfigDataClient();

const application_name = "MyDemoApp";
const environment_name = "MyEnvironment";
const config_profile_name = "MyConfigProfile";

// start a new configuration session.
// this operation does not return configuration data.
```

```
// rather, it returns an initial configuration token that should be passed to
 GetLatestConfiguration.
//
// note: this operation should only be performed once (per configuration).
//   all subsequent calls to AppConfigData should be via GetLatestConfiguration.
const scs = await appconfigdata.send(
  new StartConfigurationSessionCommand({
    ApplicationIdentifier: application_name,
    EnvironmentIdentifier: environment_name,
    ConfigurationProfileIdentifier: config_profile_name,
  })
);
const { InitialConfigurationToken } = scs;

// retrieve configuration data from the session.
// this operation returns your configuration data.
// each invocation of this operation returns a unique token that should be passed to
 the subsequent invocation.
//
// note: this operation does not always return configuration data after the first
 invocation.
//   data is only returned if the configuration has changed within AWS AppConfig
 (i.e. a deployment occurred).
//   therefore, you should cache the data returned by this call so that you can use
 it later.
const glc = await appconfigdata.send(
  new GetLatestConfigurationCommand({
    ConfigurationToken: InitialConfigurationToken,
  })
);
const config = glc.Configuration.transformToString();
```

Pulizia dell'ambiente

Se hai eseguito uno o più esempi di codice in questa sezione, ti consigliamo di utilizzare uno dei seguenti esempi per individuare ed eliminare le AWS AppConfig risorse create da tali esempi di codice. Gli esempi in questa sezione richiamano le seguenti API:

- ListApplications
- DeleteApplication
- ListEnvironments

- DeleteEnvironments
- ListConfigurationProfiles
- DeleteConfigurationProfile
- ListHostedConfigurationVersions
- DeleteHostedConfigurationVersion

Java

```
/*
    This sample provides cleanup code that deletes all the AWS AppConfig resources
 created in the samples above.

   WARNING: this code will permanently delete the given application and all of its
 sub-resources, including
   configuration profiles, hosted configuration versions, and environments. DO NOT
 run this code against
   an application that you may need in the future.
   */

   public void cleanUpDemoResources() {
       AppConfigClient appconfig = AppConfigClient.create();

       // The name of the application to delete
       // IMPORTANT: verify this name corresponds to the application you wish to
 delete
       String applicationToDelete = "MyDemoApp";


 appconfig.listApplicationsPaginator(ListApplicationsRequest.builder().build()).items().forE
 -> {
           if (app.name().equals(applicationToDelete)) {
               System.out.println("Deleting App: " + app);
               appconfig.listConfigurationProfilesPaginator(req ->
 req.applicationId(app.id())).items().forEach(cp -> {
                   System.out.println("Deleting Profile: " + cp);
                   appconfig
                       .listHostedConfigurationVersionsPaginator(req -> req
                           .applicationId(app.id())
                           .configurationProfileId(cp.id()))
                       .items()
                       .forEach(hcv -> {
```

```
                            System.out.println("Deleting HCV: " + hcv);
                            appconfig.deleteHostedConfigurationVersion(req -> req
                                .applicationId(app.id())
                                .configurationProfileId(cp.id())
                                .versionNumber(hcv.versionNumber()));
                        });
                    appconfig.deleteConfigurationProfile(req -> req
                        .applicationId(app.id())
                        .configurationProfileId(cp.id()));
                });

                appconfig.listEnvironmentsPaginator(req-
>req.applicationId(app.id())).items().forEach(env -> {
                    System.out.println("Deleting Environment: " + env);
                    appconfig.deleteEnvironment(req-
>req.applicationId(app.id()).environmentId(env.id()));
                });

                appconfig.deleteApplication(req -> req.applicationId(app.id()));
            }
        });
    }
```

Python

```
# this sample provides cleanup code that deletes all the AWS AppConfig resources
 created in the samples above.
#
# WARNING: this code will permanently delete the given application and all of its
 sub-resources, including
#   configuration profiles, hosted configuration versions, and environments. DO NOT
 run this code against
#   an application that you may need in the future.
#

import boto3

# the name of the application to delete
# IMPORTANT: verify this name corresponds to the application you wish to delete
application_name = 'MyDemoApp'

# create and iterate over a list paginator such that we end up with a list of pages,
 which are themselves lists of applications
```

```
# e.g. [ [{'Name':'MyApp1',...},{'Name':'MyApp2',...}], [{'Name':'MyApp3',...}] ]
list_of_app_lists = [page['Items'] for page in
 appconfig.get_paginator('list_applications').paginate()]
# retrieve the target application from the list of lists
application = [app for apps in list_of_app_lists for app in apps if app['Name'] ==
 application_name][0]
print(f"deleting application {application['Name']} (id={application['Id']})")

# delete all configuration profiles
list_of_config_lists = [page['Items'] for page in
 appconfig.get_paginator('list_configuration_profiles').paginate(ApplicationId=application['
for config_profile in [config for configs in list_of_config_lists for config in
 configs]:
    print(f"\tdeleting configuration profile {config_profile['Name']}
 (Id={config_profile['Id']})")

    # delete all hosted configuration versions
    list_of_hcv_lists = [page['Items'] for page in
 appconfig.get_paginator('list_hosted_configuration_versions').paginate(ApplicationId=applic
 ConfigurationProfileId=config_profile['Id'])]
    for hcv in [hcv for hcvs in list_of_hcv_lists for hcv in hcvs]:

 appconfig.delete_hosted_configuration_version(ApplicationId=application['Id'],
 ConfigurationProfileId=config_profile['Id'], VersionNumber=hcv['VersionNumber'])
        print(f"\t\tdeleted hosted configuration version {hcv['VersionNumber']}")

    # delete the config profile itself
    appconfig.delete_configuration_profile(ApplicationId=application['Id'],
 ConfigurationProfileId=config_profile['Id'])
    print(f"\tdeleted configuration profile {config_profile['Name']}
 (Id={config_profile['Id']})")

# delete all environments
list_of_env_lists = [page['Items'] for page in
 appconfig.get_paginator('list_environments').paginate(ApplicationId=application['Id'])]
for environment in [env for envs in list_of_env_lists for env in envs]:
    appconfig.delete_environment(ApplicationId=application['Id'],
 EnvironmentId=environment['Id'])
    print(f"\tdeleted environment {environment['Name']} (Id={environment['Id']})")

# delete the application itself
appconfig.delete_application(ApplicationId=application['Id'])
print(f"deleted application {application['Name']} (id={application['Id']})")
```

JavaScript

```
// this sample provides cleanup code that deletes all the AWS AppConfig resources
 created in the samples above.

// WARNING: this code will permanently delete the given application and all of its
 sub-resources, including
//   configuration profiles, hosted configuration versions, and environments. DO NOT
 run this code against
//   an application that you may need in the future.

import {
  AppConfigClient,
  paginateListApplications,
  DeleteApplicationCommand,
  paginateListConfigurationProfiles,
  DeleteConfigurationProfileCommand,
  paginateListHostedConfigurationVersions,
  DeleteHostedConfigurationVersionCommand,
  paginateListEnvironments,
  DeleteEnvironmentCommand,
} from "@aws-sdk/client-appconfig";

const client = new AppConfigClient();

// the name of the application to delete
// IMPORTANT: verify this name corresponds to the application you wish to delete
const application_name = "MyDemoApp";

// iterate over all applications, deleting ones that have the name defined above
for await (const app_page of paginateListApplications({ client }, {})) {
  for (const application of app_page.Items) {

    // skip applications that dont have the name thats set
    if (application.Name !== application_name) continue;

    console.log( `deleting application ${application.Name} (id=${application.Id})`);

    // delete all configuration profiles
    for await (const config_page of paginateListConfigurationProfiles({ client },
 { ApplicationId: application.Id })) {
      for (const config_profile of config_page.Items) {
        console.log(`\tdeleting configuration profile ${config_profile.Name} (Id=
${config_profile.Id})`);
```

```
        // delete all hosted configuration versions
        for await (const hosted_page of
 paginateListHostedConfigurationVersions({ client },
          { ApplicationId: application.Id, ConfigurationProfileId:
 config_profile.Id }
        )) {
          for (const hosted_config_version of hosted_page.Items) {
            await client.send(
              new DeleteHostedConfigurationVersionCommand({
                ApplicationId: application.Id,
                ConfigurationProfileId: config_profile.Id,
                VersionNumber: hosted_config_version.VersionNumber,
              })
            );
            console.log(`\t\tdeleted hosted configuration version
 ${hosted_config_version.VersionNumber}`);
          }
        }

        // delete the config profile itself
        await client.send(
          new DeleteConfigurationProfileCommand({
            ApplicationId: application.Id,
            ConfigurationProfileId: config_profile.Id,
          })
        );
        console.log(`\tdeleted configuration profile ${config_profile.Name} (Id=
${config_profile.Id})`)
      }

      // delete all environments
      for await (const env_page of paginateListEnvironments({ client },
 { ApplicationId: application.Id })) {
        for (const environment of env_page.Items) {
          await client.send(
            new DeleteEnvironmentCommand({
              ApplicationId: application.Id,
              EnvironmentId: environment.Id,
            })
          );
          console.log(`\tdeleted environment ${environment.Name} (Id=
${environment.Id})`)
        }
```

```
      }
    }

    // delete the application itself
    await client.send(
      new DeleteApplicationCommand({ ApplicationId: application.Id })
    );
    console.log(`deleted application ${application.Name} (id=${application.Id})`)
  }
}
```

Sicurezza in AWS AppConfig

La sicurezza del cloud in AWS ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il modello di responsabilità condivisa fa riferimento ad una sicurezza del cloud e nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue i servizi AWS in Cloud AWS. AWS fornisce inoltre i servizi che è possibile utilizzare in modo sicuro. Revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei Programmi di conformità AWS. Per informazioni sui programmi di conformità applicabili a AWS Systems Manager, consulta Servizi AWScoperti dal programma di conformità.
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

AWS AppConfigè una funzionalità diAWS Systems Manager. Per capire come applicare il modello di responsabilità condivisa durante l'utilizzoAWS AppConfig, consulta Security in AWS Systems Manager. Questa sezione descrive come configurare Systems Manager per soddisfare gli obiettivi di sicurezza e conformità perAWS AppConfig.

Implementazione dell'accesso con privilegi minimi

Come best practice di sicurezza, concedi le autorizzazioni minime richieste dalle identità per eseguire azioni specifiche su risorse specifiche in condizioni specifiche. AWS AppConfig L'agente offre due funzionalità che consentono all'agente di accedere al file system di un'istanza o contenitore: backup e scrittura su disco. Se abiliti queste funzionalità, verifica che solo l'AWS AppConfigagente disponga delle autorizzazioni di scrittura sui file di configurazione designati sul file system. Verifica inoltre che solo i processi necessari per leggere da questi file di configurazione siano in grado di farlo. L'applicazione dell'accesso con privilegio minimo è fondamentale per ridurre i rischi di sicurezza e l'impatto risultante da errori o intenzioni dannose.

Per ulteriori informazioni sull'implementazione dell'accesso con privilegi minimi, vedere SEC03-BP02 Garantire l'accesso con privilegi minimi nella Guida per l'utente. AWS Well-Architected Tool Per

ulteriori informazioni sulle funzionalità dell'AWS AppConfigagente menzionate in questa sezione, vedere. Funzionalità di recupero aggiuntive

Crittografia dei dati a riposo per AWS AppConfig

AWS AppConfigfornisce la crittografia di default per proteggere i dati inattivi dei clienti utilizzandoChiavi di proprietà di AWS.

Chiavi di proprietà di AWS— AWS AppConfig utilizza queste chiavi per impostazione predefinita per crittografare automaticamente i dati distribuiti dal servizio e ospitati nell'archivio AWS AppConfig dati. Non è possibile visualizzare, gestireChiavi di proprietà di AWS, utilizzare o controllare il loro utilizzo. Tuttavia, non è necessario effettuare alcuna operazione o modificare programmi per proteggere le chiavi che eseguono la crittografia dei dati. Per ulteriori informazioni, consulta la sezione Chiavi di proprietà di AWS nella Guida per gli sviluppatori di AWS Key Management Service.

Sebbene non sia possibile disabilitare questo livello di crittografia o selezionare un tipo di crittografia alternativo, è possibile specificare una chiave gestita dal cliente da utilizzare quando si salvano i dati di configurazione ospitati nel AWS AppConfig data store e quando si distribuiscono i dati di configurazione.

Chiavi gestite dal cliente: AWS AppConfig supporta l'uso di una chiave simmetrica gestita dal cliente che puoi creare, possedere e gestire per aggiungere un secondo livello di crittografia rispetto a quello esistente. Chiave di proprietà di AWS Avendo il pieno controllo di questo livello di crittografia, è possibile eseguire operazioni quali:

- Stabilire e mantenere politiche e sovvenzioni chiave
- Stabilire e mantenere le politiche IAM
- Abilitare e disabilitare le policy delle chiavi
- Ruotare i materiali crittografici delle chiavi
- Aggiungere tag
- Creare alias delle chiavi
- Pianificare l’eliminazione delle chiavi

Per ulteriori informazioni, consulta Customer managed key nella AWS Key Management ServiceDeveloper Guide.

AWS AppConfigsupporta chiavi gestite dal cliente

AWS AppConfigoffre supporto per la crittografia a chiave gestita dal cliente per i dati di configurazione. Per le versioni di configurazione salvate nell'archivio dati AWS AppConfig ospitato, i clienti possono impostare un profilo di configurazione `KmsKeyIdentifier` sul profilo di configurazione corrispondente. Ogni volta che viene creata una nuova versione dei dati di configurazione utilizzando l'operazione `CreateHostedConfigurationVersion` API, AWS AppConfig genera una chiave AWS KMS dati da cui `KmsKeyIdentifier` crittografare i dati prima di archiviarli. Quando successivamente si accede ai dati, durante le operazioni `GetHostedConfigurationVersion` o tramite l'`StartDeployment`API, AWS AppConfig decripta i dati di configurazione utilizzando le informazioni sulla chiave dati generata.

AWS AppConfigoffre inoltre supporto per la crittografia a chiave gestita dal cliente per i dati di configurazione distribuiti. Per crittografare i dati di configurazione, i clienti possono fornire una procedura `KmsKeyIdentifier` di implementazione. AWS AppConfiggenera la chiave AWS KMS dati con questa chiave `KmsKeyIdentifier` per crittografare i dati sul funzionamento dell'`StartDeployment`API.

AWS AppConfigaccesso alla crittografia

Quando crei una chiave gestita dal cliente, utilizza la seguente politica di chiave per assicurarti che la chiave possa essere utilizzata.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Sid": "Allow use of the key",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::account_ID:role/role_name"
        },
        "Action": [
            "kms:Decrypt",
            "kms:GenerateDataKey"
            ],
        "Resource": "*"
        }
 ]
```

Per crittografare i dati di configurazione ospitati con una chiave gestita dal cliente, la chiamata di identità `CreateHostedConfigurationVersion` richiede la seguente dichiarazione di politica, che può essere assegnata a un utente, gruppo o ruolo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:GenerateDataKey,
            "Resource": "arn:aws:kms:Region:account_ID:key_ID"
        }
    ]
}
```

Se si utilizza un segreto di Secrets Manager o qualsiasi altro dato di configurazione crittografato con una chiave gestita dal cliente, `retrievalRoleArn` sarà necessario `kms:Decrypt` decrittografare e recuperare i dati.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "arn:aws:kms:Region:account_ID:configuration source/object"
    }
  ]
}
```

Quando si chiama l'operazione AWS AppConfig StartDeploymentAPI, la chiamata di identità `StartDeployment` richiede la seguente policy IAM, che può essere assegnata a un utente, gruppo o ruolo:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*"
      ],
      "Resource": "arn:aws:kms:Region:account_ID:key_ID"
    }
  ]
```

```
}
```

Quando si chiama l'operazione AWS AppConfig GetLatestConfigurationAPI, la chiamata di identità `GetLatestConfiguration` richiede la seguente politica che può essere assegnata a un utente, gruppo o ruolo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt,
            "Resource": "arn:aws:kms:Region:account_ID:key_ID"
        }
    ]
}
```

Contesto di crittografia

Un contesto di crittografia è un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali sui dati.

AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi per supportare la crittografia autenticata. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, AWS KMS lega il contesto di crittografia ai dati crittografati. Per decrittografare i dati, nella richiesta deve essere incluso lo stesso contesto di crittografia.

AWS AppConfigcontesto di crittografia: AWS AppConfig utilizza un contesto di crittografia in tutte le operazioni AWS KMS crittografiche per i dati e le distribuzioni di configurazione ospitati crittografati. Il contesto contiene una chiave corrispondente al tipo di dati e un valore che identifica l'elemento di dati specifico.

Monitoraggio delle chiavi di crittografia per AWS

Quando utilizzi chiavi gestite AWS KMS dal cliente conAWS AppConfig, puoi utilizzare AWS CloudTrail o Amazon CloudWatch Logs per tenere traccia delle richieste AWS AppConfig inviate aAWS KMS.

L'esempio seguente è un CloudTrail evento per `Decrypt` monitorare AWS KMS le operazioni richiamate per accedere AWS AppConfig ai dati crittografati dalla chiave gestita dal cliente:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "appconfig.amazonaws.com"
    },
    "eventTime": "2023-01-03T02:22:28z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "Region",
    "sourceIPAddress": "172.12.34.56",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "encryptionContext": {
            "aws:appconfig:deployment:arn":
 "arn:aws:appconfig:Region:account_ID:application/application_ID/
environment/environment_ID/deployment/deployment_ID"
        },
        "keyId": "arn:aws:kms:Region:account_ID:key/key_ID",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "account_ID",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:Region:account_ID:key_ID"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "account_ID",
    "sharedEventID": "dc129381-1d94-49bd-b522-f56a3482d088"
}
```

Accesso AWS AppConfig tramite un'interfaccia endpoint () AWS PrivateLink

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e. AWS AppConfig Puoi accedere AWS AppConfig come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. AWS AppConfig

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Queste sono interfacce di rete gestite dal richiedente che fungono da punto di ingresso per il traffico destinato a AWS AppConfig.

Per ulteriori informazioni, consulta la sezione Accesso a Servizi AWS tramite AWS PrivateLink nella Guida di AWS PrivateLink.

Considerazioni per AWS AppConfig

Prima di configurare un endpoint di interfaccia perAWS AppConfig, consulta le considerazioni nella Guida. AWS PrivateLink

AWS AppConfigsupporta l'effettuazione di chiamate verso `appconfig`e `appconfigdata`servizi tramite l'endpoint dell'interfaccia.

Crea un endpoint dell'interfaccia per AWS AppConfig

Puoi creare un endpoint di interfaccia per AWS AppConfig utilizzare la console Amazon VPC o AWS Command Line Interface (). AWS CLI Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink.

Crea un endpoint di interfaccia per AWS AppConfig utilizzare i seguenti nomi di servizio:

```
com.amazonaws.region.appconfig
```

```
com.amazonaws.region.appconfigdata
```

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API AWS AppConfig utilizzando il nome DNS regionale predefinito. Ad esempio `appconfig.us-east-1.amazonaws.com` e `appconfigdata.us-east-1.amazonaws.com`.

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo AWS AppConfig tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito AWS AppConfig dal tuo VPC, collega una policy endpoint personalizzata all'endpoint di interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (Account AWS, utenti IAM e ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink.

Esempio: policy di endpoint VPC per le operazioni dell'AWS AppConfig

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Se collegata a un endpoint dell'interfaccia, questa policy concede l'accesso alle operazioni AWS AppConfig elencate per tutti i principali su tutte le risorse.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
         "Action": [
            "appconfig:CreateApplication",
            "appconfig:CreateEnvironment",
            "appconfig:CreateConfigurationProfile",
            "appconfig:StartDeployment",
            "appconfig:GetLatestConfiguration"
            "appconfig:StartConfigurationSession"
         ],
         "Resource":"*"
      }
   ]
}
```

Rotazione dei tasti di Secrets Manager

Questa sezione descrive importanti informazioni di sicurezza sull'AWS AppConfigintegrazione con Secrets Manager. Per informazioni su Secrets Manager, vedi Cos'èAWS Secrets Manager? nella Guida AWS Secrets Manager per l'utente.

Impostazione della rotazione automatica dei segreti di Secrets Manager distribuita da AWS AppConfig

La rotazione è il processo di aggiornamento periodico di un segreto archiviato in Secrets Manager. Quando si ruota un segreto, vengono aggiornati sia il segreto in Secrets Manager che le credenziali del database o del servizio. È possibile configurare la rotazione automatica dei segreti in Secrets Manager utilizzando una AWS Lambda funzione per aggiornare il segreto e il database. Per ulteriori informazioni, consulta Ruotare AWS Secrets Manager i segreti nella Guida per l'AWS Secrets Managerutente.

Per abilitare la rotazione delle chiavi dei segreti di Secrets Manager distribuiti daAWS AppConfig, aggiorna la funzione di rotazione Lambda e distribuisci il segreto ruotato.

> **Note**
>
> Implementa il tuo profilo di AWS AppConfig configurazione dopo che il tuo segreto è stato ruotato e completamente aggiornato alla nuova versione. Puoi determinare se il segreto è ruotato perché lo stato `VersionStage` cambia da a. `AWSPENDING` `AWSCURRENT` Il completamento della rotazione segreta avviene all'interno della `finish_secret` funzione Secrets Manager Rotation Templates.

Ecco un esempio di funzione che avvia una AWS AppConfig distribuzione dopo la rotazione di un segreto.

```
import time
import boto3
client = boto3.client('appconfig')

def finish_secret(service_client, arn, new_version):
    """Finish the rotation by marking the pending secret as current
    This method finishes the secret rotation by staging the secret staged AWSPENDING
 with the AWSCURRENT stage.
```

```
    Args:
        service_client (client): The secrets manager service client
        arn (string): The secret ARN or other identifier
        new_version (string): The new version to be associated with the secret
    """
    # First describe the secret to get the current version
    metadata = service_client.describe_secret(SecretId=arn)
    current_version = None
    for version in metadata["VersionIdsToStages"]:
        if "AWSCURRENT" in metadata["VersionIdsToStages"][version]:
            if version == new_version:
                # The correct version is already marked as current, return
                logger.info("finishSecret: Version %s already marked as AWSCURRENT for
%s" % (version, arn))
                return
            current_version = version
            break

    # Finalize by staging the secret version current
    service_client.update_secret_version_stage(SecretId=arn, VersionStage="AWSCURRENT",
MoveToVersionId=new_version, RemoveFromVersionId=current_version)

    # Deploy rotated secret
    response = client.start_deployment(
            ApplicationId='TestApp',
            EnvironmentId='TestEnvironment',
            DeploymentStrategyId='TestStrategy',
            ConfigurationProfileId='ConfigurationProfileId',
            ConfigurationVersion=new_version,
            KmsKeyIdentifier=key,
            Description='Deploy secret rotated at ' + str(time.time())
        )

    logger.info("finishSecret: Successfully set AWSCURRENT stage to version %s for
secret %s." % (new_version, arn))
```

Monitoraggio di AWS AppConfig

Il monitoraggio è importante per mantenere l'affidabilità, la disponibilità e le prestazioni di AWS AppConfig e delle altre soluzioni AWS. AWS fornisce i seguenti strumenti di monitoraggio per controllare AWS AppConfig, segnalare eventuali problemi ed eseguire operazioni automatiche quando appropriato:

- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account AWS e fornisce i file di log a un bucket Amazon S3 specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la Guida per l'utente AWS CloudTrail.
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di log da istanze Amazon EC2 e altre CloudTrail fonti. CloudWatch I log possono monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.

Argomenti

Registrazione di chiamate API AWS AppConfig con AWS CloudTrail

AWS AppConfigè integrato conAWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, da un ruolo o da un AWS servizio inAWS AppConfig. CloudTrail acquisisce tutte le chiamate API AWS AppConfig come eventi. Le chiamate acquisite includono le chiamate dalla console di AWS AppConfig e le chiamate di codice alle operazioni delle API AWS AppConfig. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS AppConfig Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviataAWS AppConfig, l'indirizzo IP

da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

AWS AppConfiginformazioni in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività inAWS AppConfig, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti in Account AWS. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nell'Account AWS che includa gli eventi per AWS AppConfig, crea un trail. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un percorso nella console, questo sarà valido in tutte le Regioni AWS. Il percorso registra gli eventi di tutte le Regioni nella partizione AWSe distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Panoramica della creazione di un percorso
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte AWS AppConfig le azioni vengono registrate CloudTrail e documentate nell'AWS AppConfigAPI Reference. Ad esempio, le chiamate a `GetApplication` e `CreateApplication` le `ListApplications` azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.

- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

AWS AppConfigeventi di dati in CloudTrail

Gli eventi relativi ai dati forniscono informazioni sulle operazioni eseguite sulle risorse su o all'interno di una risorsa (ad esempio, recuperando l'ultima configurazione distribuita tramite chiamata). GetLatestConfiguration Queste operazioni sono definite anche operazioni del piano dei dati. Gli eventi di dati sono spesso attività che interessano volumi elevati di dati. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione AWS CloudTrailPrezzi.

Puoi registrare gli eventi relativi ai dati per i tipi di AWS AppConfig risorse utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail dell'API. La tabella in questa sezione mostra i tipi di risorse disponibili perAWS AppConfig.

- Per registrare gli eventi relativi ai dati utilizzando la CloudTrail console, crea un percorso o un data store di eventi per registrare gli eventi di dati oppure aggiorna un trail o un data store di eventi esistente per registrare gli eventi di dati.

  1. Scegli Data events per registrare gli eventi relativi ai dati.
  2. Dall'elenco dei tipi di eventi Data, scegli AWS AppConfig.
  3. Scegli il modello di selettore di registro che desideri utilizzare. Puoi registrare tutti gli eventi relativi ai dati per il tipo di risorsa, registrare tutti `readOnly` gli eventi, registrare tutti `writeOnly` gli eventi o creare un modello di selettore di registro personalizzato per filtrare i `readOnly` campi`eventName`, `eresources.ARN`.
  4. Per il nome del selettore, immettere. AppConfigDataEvents Per informazioni sull'attivazione di Amazon CloudWatch Logs per il percorso degli eventi relativi ai dati, consultaMetriche di registrazione per AWS AppConfig le chiamate sul piano dati.
- Per registrare gli eventi relativi ai dati utilizzando ilAWS CLI, configura il `--advanced-event-selectors` parametro in modo che il `eventCategory` campo sia uguale `Data` e il `resources.type` campo uguale al valore del tipo di risorsa (vedi tabella). È possibile aggiungere condizioni per filtrare i valori dei `resources.ARN` campi `readOnly``eventName`, e.

- Per configurare un percorso per registrare gli eventi relativi ai dati, esegui il put-event-selectorscomando. Per ulteriori informazioni, vedere Registrazione degli eventi relativi ai dati per i AWS CLI percorsi con.
- Per configurare un Event Data Store per registrare gli eventi di dati, esegui il create-event-data-storecomando per creare un nuovo Event Data Store per registrare gli eventi di dati oppure esegui il update-event-data-storecomando per aggiornare un Event Data Store esistente. Per ulteriori informazioni, vedere Registrazione degli eventi di dati per i data store di eventi con. AWS CLI

Nella tabella seguente sono elencati i tipi di risorse AWS AppConfig. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type (console) sulla CloudTrail console. La colonna del valore resources.type mostra il `resources.type` valore da specificare durante la configurazione dei selettori di eventi avanzati utilizzando le API o. AWS CLI CloudTrail La CloudTrail colonna Data API loggate mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	API di dati registrate su* CloudTrail
AWS AppConfig	`AWS::AppConfig::Configuration`	• GetLatestConfiguration • StartConfigurationSession

*Puoi configurare selettori di eventi avanzati per filtrare in base a `eventName`, `readOnly`, e `resources.ARN` campi per registrare solo gli eventi che ritieni importanti. Per ulteriori informazioni sui campi, consulta AdvancedFieldSelector.

AWS AppConfiggestione degli eventi in CloudTrail

Gli eventi di gestione forniscono informazioni sulle operazioni di gestione eseguite sulle risorse nel tuo account AWS. Queste operazioni sono definite anche operazioni del piano di controllo. Per impostazione predefinita, CloudTrail registra gli eventi di gestione.

AWS AppConfigregistra tutte le operazioni AWS AppConfig del piano di controllo come eventi di gestione. Per un elenco delle operazioni del piano di AWS AppConfig controllo a cui si AWS AppConfig effettua l'accesso CloudTrail, consulta l'AWS AppConfigAPI Reference.

Comprensione delle voci dei file di log di AWS AppConfig

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'StartConfigurationSessionazione.


```
{
      "eventVersion": "1.09",
      "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/Administrator",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
          "sessionIssuer": {},
          "attributes": {
            "creationDate": "2024-01-11T14:37:02Z",
            "mfaAuthenticated": "false"
          }
        }
      },
      "eventTime": "2024-01-11T14:45:15Z",
      "eventSource": "appconfig.amazonaws.com",
      "eventName": "StartConfigurationSession",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "203.0.113.0",
      "userAgent": "Boto3/1.34.11 md/Botocore#1.34.11 ua/2.0 os/macos#22.6.0
 md/arch#x86_64 lang/python#3.11.4 md/pyimpl#CPython cfg/retry-mode#legacy
 Botocore/1.34.11",
      "requestParameters": {
        "applicationIdentifier": "rrfexample",
        "environmentIdentifier": "mexampleqe0",
        "configurationProfileIdentifier": "3eexampleu1"
      },
      "responseElements": null,
```

```
      "requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE",
      "readOnly": false,
      "resources": [
        {
          "accountId": "123456789012",
          "type": "AWS::AppConfig::Configuration",
          "ARN": "arn:aws:appconfig:us-east-1:123456789012:application/rrfexample/
environment/mexampleqe0/configuration/3eexampleu1"
        }
      ],
      "eventType": "AwsApiCall",
      "managementEvent": false,
      "recipientAccountId": "123456789012",
      "eventCategory": "Data",
      "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "appconfigdata.us-east-1.amazonaws.com"
      }
    }
```

Metriche di registrazione per AWS AppConfig le chiamate sul piano dati

Se hai configurato AWS CloudTrail per registrare gli eventi AWS AppConfig relativi ai dati, puoi abilitare Amazon CloudWatch Logs per registrare le metriche per le chiamate sul piano AWS AppConfig dati. Puoi quindi cercare e filtrare i dati di log in CloudWatch Logs creando uno o più filtri metrici. I filtri metrici definiscono i termini e i modelli da cercare nei dati di registro quando vengono inviati ai registri. CloudWatch CloudWatch Logs utilizza filtri metrici per trasformare i dati di registro in metriche numeriche. CloudWatch Puoi rappresentare graficamente le metriche o configurarle con un allarme.

Prima di iniziare

Abilita la registrazione degli eventi AWS AppConfig relativi ai dati in. AWS CloudTrail La procedura seguente descrive come abilitare la registrazione metrica per un trail in esistente AWS AppConfig. CloudTrail Per informazioni su come abilitare la CloudTrail registrazione per le chiamate al piano AWS AppConfig dati, vedere. AWS AppConfigeventi di dati in CloudTrail

Utilizzare la procedura seguente per consentire a CloudWatch Logs di registrare le metriche per le chiamate al AWS AppConfig piano dati.

Per consentire a CloudWatch Logs di registrare le metriche per le chiamate sul piano dati AWS AppConfig

1. Apri la CloudTrail console all'indirizzo https://console.aws.amazon.com/cloudtrail/.
2. Nella dashboard, scegli il tuo AWS AppConfig percorso.
3. Nella sezione CloudWatch Registri, scegli Modifica.
4. Scegli Enabled (Abilitato).
5. Per il nome del gruppo di log, lascia il nome predefinito o inserisci un nome. Prendere nota del nome. Sceglierai il gruppo di log nella console CloudWatch Logs in un secondo momento.
6. In Role name (Nome ruolo), immettere un nome.
7. Seleziona Salvataggio delle modifiche.

Utilizza la procedura seguente per creare una metrica e un filtro metrico per AWS AppConfig Logs. CloudWatch La procedura descrive come creare un filtro metrico per le chiamate da `operation` e (facoltativamente) le chiamate da e. `operation` `Amazon Resource Name (ARN)`

Per creare una metrica e un filtro metrico per in Logs AWS AppConfig CloudWatch

1. Apri la console all' CloudWatch indirizzo. https://console.aws.amazon.com/cloudwatch/
2. Nel pannello di navigazione scegli Log, quindi Gruppi di log.
3. Scegli la casella di controllo accanto al gruppo di AWS AppConfig log.
4. Scegli Actions (Operazioni) e quindi Create metric filter (Crea filtro parametri).
5. Per Nome del filtro, inserisci un nome.
6. Per Pattern di filtro, inserisci quanto segue:

   ```
   { $.eventSource = "appconfig.amazonaws.com" }
   ```

7. (Facoltativo) Nella sezione Schema di test, scegliete il gruppo di log dall'elenco Seleziona i dati di registro da testare. Se CloudTrail non ha registrato alcuna chiamata, puoi saltare questo passaggio.
8. Seleziona Avanti.
9. Per Metric namespace, inserisci. **`AWS AppConfig`**

10. Per Nome parametro, inserire **`Calls`**.
11. In Metric value (Valore parametro), inserisci **`1`**.
12. Ignora il valore e l'unità predefiniti.
13. Per Nome dimensione, immettere**`operation`**.
14. Per Valore della dimensione, immettere**`$.eventName`**.

    (Facoltativo) Puoi inserire una seconda dimensione che includa l'Amazon Resource Name (ARN) che effettua la chiamata. Per aggiungere una seconda dimensione, in Nome dimensione, inserisci**`resource`**. Per Valore della dimensione, immettere**`$.resources[0].ARN`**.

    Seleziona Avanti.
15. Controlla i dettagli del filtro e crea un filtro metrico.

(Facoltativo) Puoi ripetere questa procedura per creare un nuovo filtro metrico per un codice di errore specifico come. AccessDenied Se lo fai, inserisci i seguenti dettagli:

1. Per Nome filtro, inserisci un nome.
2. Per Pattern di filtro, inserisci quanto segue:

   ```
   { $.errorCode = "codename" }
   ```

   Ad esempio

   ```
   { $.errorCode = "AccessDenied" }
   ```
3. Per Metric namespace, immettere. **`AWS AppConfig`**
4. Per Nome parametro, inserire **`Errors`**.
5. In Metric value (Valore parametro), inserisci **`1`**.
6. Per Valore predefinito, inserite uno zero (0).
7. Ignora unità, dimensioni e allarmi.

Dopo aver CloudTrail registrato le chiamate API, puoi visualizzare le metriche in. CloudWatch Per ulteriori informazioni, consulta Visualizzazione delle metriche e dei log nella console nella Amazon CloudWatch User Guide. Per informazioni su come individuare una metrica che hai creato, consulta Cerca metriche disponibili.

> **Note**
>
> Se imposti la metrica di errore senza dimensione, come descritto qui, puoi visualizzare tali metriche nella pagina Metriche senza dimensione.

Creazione di un allarme per una metrica CloudWatch

Dopo aver creato le metriche, puoi creare allarmi metrici in. CloudWatch Ad esempio, puoi creare un allarme per la metrica delle AWS AppConfigchiamate creata nella procedura precedente. In particolare, è possibile creare un allarme per le chiamate all'azione dell'AWS AppConfig`StartConfigurationSession`API che superano una soglia. Per informazioni su come creare un allarme per una metrica, consulta Creare un CloudWatch allarme basato su una soglia statica nella Amazon CloudWatch User Guide. Per informazioni sui limiti predefiniti per le chiamate al piano AWS AppConfig dati, consulta Limiti predefiniti del piano dati nel Riferimenti generali di Amazon Web Services.

AWS AppConfig Cronologia dei documenti della Guida dell'utente

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di. AWS AppConfig

Versione attuale dell'API: 2019-10-09

Modifica	Descrizione	Data
AWS AppConfig esempi di estensioni personalizzate	L'argomento Procedura dettagliata: creazione di AWS AppConfig estensioni personalizzate ora include collegamenti alle seguenti estensioni di esempio su: GitHub • Estensione di esempio che impedisce le implement azioni con un calendario di `blocked day` moratoria utilizzando Systems Manager Change Calendar • Estensione di esempio che impedisce la divulgazione di segreti nei dati di configura zione utilizzando git-secrets • Estensione di esempio che impedisce la fuoriusci ta di informazioni di identificazione personale (PII) nei dati di configura zione utilizzando Amazon Comprehend	28 febbraio 2024

Nuovo argomento: registrazione delle chiamate AWS AppConfig API utilizzando AWS CloudTrail	AWS AppConfig è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in AWS AppConfig. CloudTrail acquisisce tutte le chiamate API AWS AppConfig come eventi. Questo nuovo argomento fornisce contenuti AWS AppConfig specifici anziché collegamenti al contenuto corrispondente nella Guida per l'utente. AWS Systems Manager Per ulteriori informazioni, consulta Registrazione delle chiamate AWS AppConfig API utilizzando. AWS CloudTrail	18 gennaio 2024

AWS AppConfig ora supporta AWS PrivateLink	Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e. AWS AppConfig Puoi accedere AWS AppConfig come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. AWS AppConfig Per ulteriori informazioni, consulta Accedere AWS AppConfig utilizzando un endpoint di interfaccia ().AWS PrivateLink	6 dicembre 2023

Funzionalità aggiuntive di recupero degli AWS AppConfig agenti e una nuova modalità di sviluppo locale	AWS AppConfig Agent offre le seguenti funzionalità aggiuntive per aiutarvi a recuperare le configurazioni per le vostre applicazioni. Funzionalità di recupero aggiuntive • Recupero di più account: utilizza AWS AppConfig Agent da un sistema primario o di recupero Account AWS per recuperare i dati di configurazione da più account fornitore. • Scrittura della copia della configurazione su disco: utilizza AWS AppConfig Agent per scrivere i dati di configurazione su disco. Questa funzionalità consente l'integrazione con i clienti con applicazioni che leggono i dati di configurazione dal disco AWS AppConfig. **Note** La configurazione di scrittura su disco non è progettata come funzionalità di backup della configurazione. AWS	1 dicembre 2023

AppConfig L'agente non legge i file di configurazione copiati su disco. Se desideri eseguire il backup delle configurazioni su disco, consulta le `BACKUP_DIRECTORY` variabili di `PRELOAD_BACKUP` ambiente per Using AWS AppConfig Agent with Amazon EC2 o AWS AppConfig Using Agent with Amazon ECS e Amazon EKS.

Modalità di sviluppo locale

AWS AppConfig L'agente supporta una modalità di sviluppo locale. Se si abilita la modalità di sviluppo locale, l'agente legge i dati di configurazione da una directory specificata su disco. Non recupera i dati di configurazione da. AWS AppConfigÈ possibile simulare le distribuzioni di configura zione aggiornando i file nella directory specificata. Consiglia mo la modalità di sviluppo locale per i seguenti casi d'uso:

	• Prova diverse versioni di configurazione prima di distribuirle utilizzando AWS AppConfig. • Prova diverse opzioni di configurazione per una nuova funzionalità prima di apportare modifiche al tuo repository di codice. • Prova diversi scenari di configurazione per verificar e che funzionino come previsto.	
Nuovo argomento relativo agli esempi di codice	È stato aggiunto un nuovo argomento sugli esempi di codice a questa guida. L'argomento include esempi in Java, Python e JavaScrip t per eseguire a livello di codice sei azioni comuni. AWS AppConfig	17 novembre 2023
Sommario rivisto per riflettere meglio il flusso di lavoro AWS AppConfig	Il contenuto di questa guida per l'utente è ora raggruppa to sotto i titoli Creazione, distribuzione, recupero ed estensione dei flussi di lavoro. Questa organizzazione riflette meglio il flusso di lavoro per l'utilizzo AWS AppConfig e mira a rendere i contenuti più reperibili.	7 novembre 2023

È stato aggiunto il riferimento al payload	L'argomento Creazione di una funzione Lambda per un' AWS AppConfig estensione personalizzata ora include un riferimento al payload di richiesta e risposta.	7 novembre 2023
Nuova strategia di AWS implementazione predefinita	AWS AppConfig ora offre e consiglia la strategia di implementazione `AppConfig.Linear20PercentEvery6Minutes` predefinita. Per ulteriori informazioni, consulta Strategie di distribuzione predefinite.	11 agosto 2023
AWS AppConfig integrazione con Amazon EC2	Puoi integrarti AWS AppConfig con le applicazioni in esecuzione sulle tue istanze Amazon Elastic Compute Cloud (Amazon EC2) Linux utilizzando Agent. AWS AppConfig L'agente supporta le architetture x86_64 e ARM64 per Amazon EC2. Per ulteriori informazioni, consulta AWS AppConfig Integrazione con Amazon EC2.	20 luglio 2023

AWS CloudFormation supporto per nuove AWS AppConfig risorse e un esempio di feature flag	AWS CloudFormation ora supporta le AWS::AppConfig::ExtensionAssociation risorse AWS::AppConfig::Extensione le risorse per aiutarti a iniziare con AWS AppConfig le estensioni. Le risorse AWS::AppConfig::ConfigurationProfile and AWS::AppConfig::HostedConfigurationVersion ora includono un esempio di creazione di un profilo di configurazione Feature Flag nell'archivio di configurazione AWS AppConfig ospitato.	12 aprile 2023

AWS AppConfig integrazione con AWS Secrets Manager	AWS AppConfig si integra con AWS Secrets Manager. Secrets Manager consente di crittografare, archiviare e recuperare in modo sicuro le credenziali per i database e altri servizi. Invece di inserire le credenziali nelle app, puoi effettuare chiamate a Secrets Manager per recuperare le credenziali ogni volta che ne hai bisogno. Secrets Manager ti aiuta a proteggere l'accesso alle tue risorse e ai tuoi dati IT consentendoti di ruotare e gestire l'accesso ai tuoi segreti. Quando si crea un profilo di configurazione in formato libero, è possibile scegliere Secrets Manager come origine dei dati di configurazione. È necessario effettuare l'onboarding con Secrets Manager e creare un segreto prima di creare il profilo di configurazione. Per ulteriori informazioni su Secrets Manager, vedi Cos'è AWS Secrets Manager? nella Guida AWS Secrets Manager per l'utente. Per informazioni sulla creazione di un profilo di configurazione, vedere Creazione di un profilo	2 febbraio 2023

di configurazione in formato libero.

AWS AppConfig integrazione con Amazon ECS e Amazon EKS	Puoi effettuare l'integrazione AWS AppConfig con Amazon Elastic Container Service (Amazon ECS) e Amazon Elastic Kubernetes Service (Amazon EKS) utilizzando l'agente. AWS AppConfig L'agente funziona come un contenitore secondario che funziona insieme alle applicazioni container Amazon ECS e Amazon EKS. L'agente migliora l'elaborazione e la gestione delle applicazioni containerizzate nei seguenti modi: • L'agente chiama AWS AppConfig per conto dell'utente utilizzando un ruolo AWS Identity and Access Management (IAM) e gestendo una cache locale di dati di configurazione. Estraendo i dati di configurazione dalla cache locale, l'applicazione richiede meno aggiornamenti del codice per gestire i dati di configurazione, recupera i dati di configurazione in millisecondi e non è interessata da problemi di rete che possono interrompere le chiamate per tali dati.	2 dicembre 2022

- L'agente offre un'esperienza nativa per il recupero e la risoluzione dei flag di funzionalità. AWS AppConfig
- Immediatamente, l'agente fornisce le migliori pratiche per le strategie di memorizzazione nella cache, gli intervalli di polling e la disponibilità dei dati di configurazione locali, tenendo traccia dei token di configurazione necessari per le successive chiamate di servizio.
- Durante l'esecuzione in background, l'agente analizza periodicamente il piano dati per verificare la presenza di aggiornamenti dei AWS AppConfig dati di configurazione. L'applicazione containerizzata può recuperare i dati connettendosi a localhost sulla porta 2772 (un valore di porta predefinito personalizzabile) e chiamando HTTP GET per recuperare i dati.
- L' AWS AppConfig agente aggiorna i dati di configurazione nei contenitori senza dover riavviare o riciclare tali contenitori.

Per ulteriori informazioni, consulta AWS AppConfig l'integrazione con Amazon ECS e Amazon EKS.

Nuova estensione: AWS AppConfig estensione per CloudWatch Evidently	Puoi usare Amazon CloudWatch Evidently per convalidare nuove funzionalità in modo sicuro offrendole e disponibili a una percentuale specifica di utenti durante il rollout della funzionalità. È possibile monitorare le prestazioni della nuova funzionalità per aiutarti a decidere quando aumentare il traffico verso gli utenti. Ciò consente di ridurre i rischi e identificare le conseguenze non intenzionali prima di avviare completamente la funzionalità. È inoltre possibile condurre esperimenti A/B per prendere decisioni sulla progettazione delle caratteristiche basate su prove e dati. L' AWS AppConfig estensione per CloudWatch Evidently consente all'applicazione di assegnare variazioni alle sessioni utente localmente anziché richiamare l'operazione. EvaluateFeature Una sessione locale mitiga i rischi di latenza e disponibilità associati a una chiamata API. Per informazioni su come configurare e utilizzare l'estensione, consulta Esegui lanci ed esperimenti A/B con	13 settembre 2022

	CloudWatch Evidently nella Amazon CloudWatch User Guide.	
Deprecazione dell'azione dell'API `GetConfiguration`	Il 18 novembre 2021, AWS AppConfig ha rilasciato un nuovo servizio di piano dati. Questo servizio sostituisce il precedente processo di recupero dei dati di configurazione utilizzando l'azione API. `GetConfiguration` Il servizio data plane utilizza due nuove azioni API e. StartConfigurationSessionGetLatestConfiguration Il servizio data plane utilizza anche nuovi endpoint. Per ulteriori informazioni, vedere Informazioni sul servizio AWS AppConfig Data Plane.	13 settembre 2022
Nuova versione dell'estensione AWS AppConfig Agent Lambda	La versione 2.0.122 dell'estensione AWS AppConfig Agent Lambda è ora disponibile. La nuova estensione utilizza diversi Amazon Resource Names (ARN). Per ulteriori informazioni, vedere le note di rilascio dell'estensione AWS AppConfig Agent Lambda.	23 agosto 2022

Lancio delle estensioni AWS AppConfig	Un'estensione aumenta la capacità di inserire logica o comportamento in punti diversi durante il AWS AppConfig flusso di lavoro di creazione o distribuzione di una configura zione. Puoi usare estensioni AWS create da te o crearne di tue. Per ulteriori informazioni, consulta Lavorare con AWS AppConfig le estensioni.	12 luglio 2022
Nuova versione dell'este nsione AWS AppConfig Agent Lambda	La versione 2.0.58 dell'este nsione AWS AppConfig Agent Lambda è ora disponibile. La nuova estensione utilizza diversi Amazon Resource Names (ARN). Per ulteriori informazioni, consulta Versioni disponibili dell'estensione AWS AppConfig Lambda.	3 maggio 2022

AWS AppConfig integrazione con Atlassian Jira	L'integrazione con Atlassian Jira consente di AWS AppConfig creare e aggiornare problemi nella console Atlassian ogni volta che si apportano modifiche a un feature flag nel modulo specificato. Account AWS Regione AWS Ogni numero di Jira include il nome del flag, l'ID dell'applicazione, l'ID del profilo di configurazione e i valori dei flag. Dopo aver aggiornato, salvato e distribuito le modifiche ai flag, Jira aggiorna i problemi esistenti con i dettagli della modifica. Per ulteriori informazioni, consulta AWS AppConfig Integrazione con Atlassian Jira.	7 aprile 2022

Disponibilità generale dei flag di funzionalità e del supporto dell'estensione Lambda per i processori ARM64 (Graviton2)	Con i flag di AWS AppConfig funzionalità, puoi sviluppar e una nuova funzionalità e distribuirla in produzione nascondendola agli utenti. Iniziate aggiungendo il flag ai dati di configurazione AWS AppConfig . Una volta che la funzionalità è pronta per essere rilasciata, puoi aggiornare i dati di configura zione del flag senza distribui re alcun codice. Questa funzionalità migliora la sicurezza dell'ambiente dev-ops perché non è necessari o distribuire nuovo codice per rilasciare la funzionalità. Per ulteriori informazioni, consulta Creazione di un profilo di configurazione del feature flag. La disponibilità generale dei flag di funzionalità AWS AppConfig include i seguenti miglioramenti: • La console include un'opzione per designare una bandiera come bandiera a breve termine. È possibile filtrare e ordinare l'elenco delle bandiere in base alle bandiere a breve termine. • Per i clienti che utilizzano i flag di funzionalità in AWS Lambda, la nuova estension	15 marzo 2022

	e Lambda consente di richiamare i singoli flag di funzionalità utilizzando un endpoint HTTP. Per ulteriori informazioni, consulta Recupero di uno o più flag da una configurazione di feature flag. Questo aggiornamento fornisce anche il supporto per AWS Lambda le estension i sviluppate per i processor i ARM64 (Graviton2). Per ulteriori informazioni, consulta Versioni disponibili dell'estensione AWS AppConfig Lambda.	
L'azione GetConfiguration API è obsoleta	L'azione `GetConfiguration` API è obsoleta. Le chiamate per ricevere i dati di configurazione devono invece utilizzare le API `StartConfigurationSession` and`GetLatestConfiguration` . Per ulteriori informazioni su queste API e su come utilizzarle, consulta Recupero della configurazione.	28 gennaio 2022

Nuova regione ARN per l'estensione Lambda AWS AppConfig	AWS AppConfig L'estensione Lambda è disponibile nella nuova regione Asia Pacifico (Osaka). L'Amazon Resource Name (ARN) è necessario per creare una Lambda nella regione. Per ulteriori informazioni sull'ARN della regione Asia Pacifico (Osaka), vedi Aggiungere l'estensione AWS AppConfig Lambda.	4 marzo 2021
AWS AppConfig Estensione Lambda	Se utilizzi AWS AppConfig per gestire le configurazioni per una funzione Lambda, ti consigliamo di aggiungere l'estensione AWS AppConfig Lambda. Questa estensione include le migliori pratiche che semplificano l'utilizzo riducendo al AWS AppConfig contempo i costi. I costi ridotti derivano da un minor numero di chiamate API al AWS AppConfig servizio e, separatamente, dalla riduzione dei costi grazie ai tempi di elaborazione delle funzioni Lambda più brevi. Per ulteriori informazioni, consulta AWS AppConfig Integrazione con le estensioni Lambda.	8 ottobre 2020

Nuova sezione	È stata aggiunta una nuova sezione che fornisce istruzioni per la configurazione AWS AppConfig. Per ulteriori informazioni, vedere Configurazione AWS AppConfig.	30 settembre 2020
Sono state aggiunte procedure da riga di comando	Le procedure di questa guida per l'utente ora includono i passaggi della riga di comando per AWS Command Line Interface (AWS CLI) e Tools for Windows. PowerShell Per ulteriori informazioni, vedere Lavorare con. AWS AppConfig	30 settembre 2020
Lancio della guida AWS AppConfig per l'utente	Utilizza AWS AppConfig, una funzionalità di AWS Systems Manager, per creare, gestire e distribuire rapidamente configurazioni di applicazioni. AWS AppConfig supporta implementazioni controllate su applicazioni di qualsiasi dimensione e include controlli di convalida e monitoraggio integrati. Puoi utilizzarlo AWS AppConfig con applicazioni ospitate su istanze EC2, contenitori AWS Lambda, applicazioni mobili o dispositivi IoT.	31 luglio 2020

Glossario AWS

Per la terminologia AWS più recente, consultare il glossario AWS nella documentazione di riferimento per Glossario AWS.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.