



Guida per l'utente

# AWS Modernizzazione del mainframe



# AWS Modernizzazione del mainframe: Guida per l'utente

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

# Table of Contents

Cos'è la modernizzazione del AWS mainframe? .....	1
Caratteristiche AWS della modernizzazione del mainframe .....	2
Modelli .....	3
Come iniziare con la modernizzazione del mainframe AWS .....	3
Servizi correlati .....	4
Accesso alla modernizzazione del AWS mainframe .....	5
Sei un utente di Mainframe Modernization per la prima volta? AWS .....	5
Prezzi .....	5
Configurazione della modernizzazione del AWS mainframe .....	6
Registrati per un Account AWS .....	6
Creazione di un utente amministratore .....	6
Nozioni di base .....	8
Tutorial: Managed Runtime per AWS Blu Age .....	8
Prerequisiti .....	9
Passaggio 1: carica l'applicazione demo .....	9
Fase 2: Creare la definizione dell'applicazione .....	9
Fase 3: Creare un ambiente di runtime .....	10
Fase 4: Creare un'applicazione .....	15
Fase 5: Implementazione di un'applicazione .....	17
Fase 6: Avviare un'applicazione .....	20
Passaggio 7: Accedere all'applicazione .....	20
Fase 8: test dell'applicazione .....	21
Pulizia delle risorse .....	22
Tutorial: runtime gestito per Micro Focus .....	22
Prerequisiti .....	23
Fase 1: creare e caricare un bucket Amazon S3 .....	24
Passaggio 2: creare e configurare un database .....	25
Passaggio 3: creare e configurare un AWS KMS key .....	27
Passaggio 4: Creare e configurare un AWS Secrets Manager database segreto .....	28
Fase 5: Creare un ambiente di runtime .....	30
Fase 6: Creare un'applicazione .....	36
Fase 7: Implementazione di un'applicazione .....	42
Fase 8: Importazione dei set di dati .....	44
Fase 9: Avviare un'applicazione .....	50

Fase 10: Connect all'applicazione CardDemo CICS .....	51
Pulizia delle risorse .....	58
Passaggi successivi .....	59
Approccio alla modernizzazione .....	60
Fase di valutazione .....	60
Fase di mobilitazione .....	60
Fase di migrazione e modernizzazione .....	61
Fase operativa e di ottimizzazione .....	61
Concetti .....	62
Applicazione .....	62
Definizione dell'applicazione .....	62
Lavoro in batch .....	63
Configurazione .....	64
Set di dati .....	64
Ambiente .....	64
Modernizzazione del mainframe .....	64
Viaggio di migrazione .....	64
Punto di montaggio .....	64
Rifattorizzazione automatizzata .....	65
Ridefinizione della piattaforma .....	65
Risorsa .....	65
Motore di runtime .....	65
AWS Blu Age Refactoring .....	66
AWS Note di rilascio di Blu Age .....	67
Note di rilascio 3.10.0 .....	68
Versione di runtime 3.10.0 .....	68
Versione 3.10.0 degli strumenti di modernizzazione .....	71
Note di rilascio 3.9.0 .....	72
Versione di runtime 3.9.0 .....	73
Versione 3.9.0 degli strumenti di modernizzazione .....	78
Note di rilascio 3.8.0 .....	81
Versione di runtime 3.8.0 .....	81
Versione 3.8.0 degli strumenti di modernizzazione .....	85
Note di rilascio 3.7.0 .....	87
Versione di runtime 3.7.0 .....	87
Versione 3.7.0 degli strumenti di modernizzazione .....	90



Note di rilascio 3.6.0 .....	92
Versione di runtime 3.6.0 .....	93
Versione 3.6.0 degli strumenti di modernizzazione .....	96
Note di rilascio 3.5.0 .....	99
Versione di runtime 3.5.0 .....	99
Versione 3.5.0 degli strumenti di modernizzazione .....	102
AWS I concetti di Blu Age Runtime .....	105
Architettura di alto livello .....	105
Struttura applicativa modernizzata .....	110
Semplificatore di dati .....	146
AWS Configurazione di Blu Age Runtime .....	154
Nozioni di base sulla configurazione delle applicazioni .....	154
Precedenza delle applicazioni .....	156
JNDI per database .....	156
Usare AWS i segreti .....	157
Altri file (groovy, sql, ecc.) .....	160
Applicazione web aggiuntiva .....	161
Abilitazione delle proprietà .....	161
Configurare l'autenticazione per le applicazioni Gapwalk .....	201
AWS API Blu Age Runtime .....	205
Creazione di URL .....	206
Applicazione Gapwalk .....	206
Endpoint REST della Blusam Application Console .....	226
Console applicativa JICS .....	243
Strutture dati .....	261
AWS Configurazione Blu Age Runtime (non gestita) .....	269
AWS Prerequisiti di Blu Age Runtime .....	269
AWS Onboarding di Blu Age Runtime .....	270
Requisiti di configurazione dell'infrastruttura .....	274
Distribuzione su Amazon ECS gestita da AWS Fargate .....	279
Distribuzione su Amazon EC2 .....	290
Modifica il codice sorgente con Blu Age Developer IDE .....	304
Tutorial: Configurazione AppStream 2.0 per AWS Blu Age Developer IDE .....	304
Tutorial: Usa AWS Blu Age Developer su AppStream 2.0 .....	310
Micro Focus Replatforming .....	327
Configurazione di Micro Focus Runtime (su Amazon EC2) .....	327

Prerequisiti .....	328
Crea l'endpoint Amazon VPC per Amazon S3 .....	328
Richiedi l'aggiornamento della lista consentita per l'account .....	330
Creazione del AWS Identity and Access Management ruolo .....	331
Concedi a License Manager le autorizzazioni richieste .....	338
Abbonati ad Amazon Machine Images .....	339
Avvia un'istanza Micro Focus per la AWS modernizzazione del mainframe .....	342
Subnet o VPC senza accesso a Internet .....	349
Risoluzione dei problemi relativi alla licenza .....	355
Tutorial: Configurazione AppStream 2.0 per Enterprise Analyzer ed Enterprise Developer .....	357
Prerequisiti .....	358
Passaggio 1: ottieni le immagini AppStream 2.0 .....	359
Passaggio 2: crea lo stack utilizzando il modello AWS CloudFormation .....	359
Fase 3: Creare un utente nella AppStream versione 2.0 .....	362
Fase 4: Accedere alla AppStream versione 2.0 .....	363
Passaggio 5: verifica i bucket in Amazon S3 (opzionale) .....	365
Fasi successive .....	365
Pulizia delle risorse .....	366
Configurare Enterprise Analyzer .....	366
Contenuto dell'immagine .....	367
Prerequisiti .....	368
Fase 1: configurazione .....	369
Passaggio 2: creare la cartella virtuale basata su Amazon S3 su Windows .....	369
Fase 3: creare una fonte ODBC per l'istanza Amazon RDS .....	370
Sessioni successive .....	372
Risoluzione dei problemi di connessione all'area di lavoro .....	372
Pulizia delle risorse .....	377
Configurare Enterprise Developer .....	377
Contenuti dell'immagine .....	378
Prerequisiti .....	379
Fase 1: Configurazione da parte dei singoli utenti Enterprise Developer .....	379
Fase 2: creare la cartella virtuale basata su Amazon S3 su Windows (opzionale) .....	380
Passaggio 3: clonare il repository .....	381
Sessioni successive .....	382
Pulizia delle risorse .....	382
Configurare l'automazione AppStream 2.0 .....	383

Configura l'automazione all'avvio della sessione .....	383
Configura l'automazione al termine della sessione .....	384
Visualizza i set di dati come tabelle .....	384
Prerequisiti .....	385
Fase 1: Configurazione della connessione ODBC al datastore Micro Focus (database Amazon RDS) .....	385
Fase 2: Creare il file MFDBFH.cfg .....	387
Fase 3: Create un file di struttura (STR) per il layout del quaderno .....	388
Fase 4: Creare una visualizzazione del database utilizzando il file di struttura (STR) .....	390
Fase 5: Visualizzate i set di dati Micro Focus come tabelle e colonne .....	391
Utilizzare i modelli con Enterprise Developer .....	392
Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine ...	392
Caso d'uso 2: utilizzo del modello di progetto COBOL senza componenti di origine .....	395
Caso d'uso 3 - Utilizzo del progetto COBOL predefinito per il collegamento alle cartelle di origine .....	397
Utilizzo del modello JSON di definizione della regione .....	399
Tutorial: configura la build per l' BankDemoesempio .....	402
Prerequisiti .....	403
Fase 1: creare bucket Amazon S3 .....	404
Passo 2: Crea il file delle specifiche di build .....	404
Passaggio 3: carica i file sorgente .....	406
Fase 4: Creare policy IAM .....	406
Fase 5: Creare un ruolo IAM .....	408
Fase 6: collegare le policy IAM al ruolo IAM .....	409
Fase 7: Creare il progetto CodeBuild .....	410
Fase 8: Avvia la compilazione .....	410
Passaggio 9: scarica gli artefatti di output .....	411
Pulizia delle risorse .....	411
Tutorial: configurazione di una pipeline CI/CD da utilizzare con Micro Focus Enterprise Developer .....	412
Prerequisiti .....	413
Crea l'infrastruttura di base della pipeline CI/CD .....	414
Crea un repository e una pipeline CI/CD AWS CodeCommit .....	418
Creazione di AppStream Enterprise Developer 2.0 .....	423
Configurazione e test di Enterprise Developer .....	423
Esercizio 1: Migliora il calcolo del prestito nell'applicazione BANKDEMO .....	428

Esercizio 2: Estrarre il calcolo del prestito nella domanda BankDemo .....	433
Pulizia delle risorse .....	436
Utilità Batch .....	437
Ubicazione binaria .....	438
Utilità M2SFTP Batch .....	438
Utilità M2WAIT Batch .....	445
Utilità TXT2PDF Batch .....	447
Utilità M2DFUTIL Batch .....	453
Utilità M2RUNCMD Batch .....	460
Replica dei dati con Precisly .....	464
Prerequisiti .....	464
Iscriviti all'Amazon Machine Image .....	464
Avvia la replica dei dati di AWS Mainframe Modernization con Precisly .....	465
Creazione di una policy IAM .....	466
Creazione di un ruolo IAM .....	467
Collega il ruolo IAM all'istanza Amazon EC2 .....	467
Integrazione con Charon .....	468
Introduzione a Charon-SSP .....	468
Sistemi operativi guest supportati .....	470
Prerequisiti dell'istanza cloud Charon-SSP .....	471
Prerequisiti dell'istanza .....	472
Creazione e configurazione di un'istanza AWS cloud per Charon (Nuova GUI) .....	473
Prerequisiti generali .....	473
Utilizzo di AWS Management Console per avviare una nuova istanza .....	475
Ripiattaforma con NTT DATA .....	480
Prerequisiti .....	480
Iscriviti all'Amazon Machine Image .....	480
Avvia la ripiattaforma AWS Mainframe Modernization con l'istanza NTT DATA .....	481
Guida introduttiva a NTT Data .....	481
Applicazioni .....	484
Creazione di un'applicazione .....	485
Creazione di un'applicazione .....	485
Implementa un'applicazione .....	486
Implementa un'applicazione .....	486
Aggiornamento di un'applicazione .....	487
Aggiornamento di un'applicazione .....	487

Eliminare un'applicazione da un ambiente .....	488
Eliminare un'applicazione da un ambiente .....	489
Eliminazione di un'applicazione .....	489
Eliminazione di un'applicazione .....	489
Invia lavori in batch per le candidature .....	490
Invia un lavoro in batch .....	491
Importazione di set di dati per le applicazioni .....	491
Importa un set di dati .....	492
Gestisci le transazioni per le applicazioni .....	492
Gestisci le transazioni per le applicazioni .....	493
Crea AWS risorse per un'applicazione migrata .....	494
Autorizzazioni richieste .....	495
Bucket Amazon S3 .....	495
Database .....	496
Chiave AWS Key Management Service .....	496
AWS Secrets Managersegreto .....	497
Configurare l'applicazione gestita .....	497
Struttura delle applicazioni gestite da AWS Blu Age .....	498
Configurazione dell'accesso alle utilità per le applicazioni gestite .....	499
Configura proprietà aggiuntive .....	508
Riferimento alla definizione dell'applicazione .....	529
Sezione generale dell'intestazione .....	529
Panoramica della sezione relativa alle definizioni .....	531
AWS Esempio di definizione dell'applicazione Blu Age .....	531
AWS Dettagli sulla definizione di Blu Age .....	532
Definizione dell'applicazione Micro Focus .....	536
Dettagli sulla definizione di Micro Focus .....	537
Riferimento alla definizione del set di dati .....	543
Proprietà comuni .....	544
Esempio di formato di richiesta di set di dati per VSAM .....	546
Esempio di formato di richiesta del set di dati per GDG Base .....	548
Esempio di formato di richiesta del set di dati per PS o GDG Generations .....	549
Esempio di formato di richiesta del set di dati per PO .....	550
Ambienti di runtime gestiti .....	553
Creare un ambiente di runtime .....	553
Crea un ambiente di runtime .....	554

Aggiornare un ambiente di runtime .....	556
Aggiorna un ambiente di runtime .....	556
Maintenance window (Finestra di manutenzione) .....	557
Arresta un ambiente di runtime .....	559
Arresta un ambiente di runtime .....	559
Riavviare un ambiente di runtime .....	560
Riavvia un ambiente di runtime .....	560
Eliminare un ambiente di runtime .....	561
Eliminare un ambiente di runtime .....	561
Test dell'applicazione .....	563
Che cos'è il test delle applicazioni? .....	563
Sei un utente di Application Testing per la prima volta? .....	564
Vantaggi del test delle applicazioni .....	564
Integrazione con AWS CloudFormation .....	565
Come funziona il test delle applicazioni .....	565
Servizi correlati .....	4
Accesso ai test delle applicazioni .....	567
Prezzi per il test delle applicazioni .....	567
Concetti di test delle applicazioni .....	568
Caso di test .....	569
Scenario di test .....	569
Progetto di test .....	570
Condizione iniziale .....	570
Registrazione (acquisizione) .....	570
Riproduci di nuovo .....	570
Confronta .....	571
Confronti tra database .....	571
Confronti tra set di dati .....	571
Stato del confronto .....	572
Regole di equivalenza .....	572
Confronto tra set di dati dello stato finale .....	573
Confronti tra database State-Progress .....	573
Equivalenza funzionale (FE) .....	573
Confronti tra schermate 3270 online .....	573
Registrazione .....	573
Riproduci i dati .....	574

Dati di riferimento .....	574
Registra, riproduci e confronta .....	574
Differenze .....	575
Equivalenze .....	575
Applicazione di origine .....	575
Applicazione di destinazione .....	575
Tutorial: Configurazione CardDemo .....	576
Prerequisiti .....	576
Fase 1: Prepararsi alla configurazione CardDemo .....	576
Fase 2: Creare tutte le risorse necessarie .....	577
Fase 3: Implementazione e avvio dell'applicazione .....	578
Fase 4: Importazione dei dati iniziali .....	578
Fase 5: Connect all' CardDemoapplicazione .....	579
Tutorial: Riproduci e confronta su Blu Age usando AWS CardDemo .....	580
Fase 1: Ottenere una Amazon Machine Image (AMI) di Amazon EC2 AWS Blu Age .....	581
Fase 2: Avvia un'istanza Amazon EC2 utilizzando l'AMI AWS Blu Age .....	581
Passaggio 3: carica i file CardDemo dipendenti su S3 .....	582
Fase 4: Caricare i database e inizializzare l'applicazione CardDemo .....	583
Fase 5: Avvia AWS il runtime Blu Age CloudFormation .....	585
Fase 6: Test dell'AWSistanza Amazon EC2 Blue Age .....	588
Passaggio 7: verifica che i passaggi precedenti siano stati completati correttamente .....	589
Fase 8. Crea una condizione iniziale .....	589
Fase 9: Creare il test case .....	590
Fase 10: Creare uno scenario di test .....	590
Fase 11: Registra lo scenario di test .....	591
Passaggio 12: Riproduci e confronta .....	591
Pagine di codici relative ai set di dati supportati .....	592
Trasferimento di file .....	603
Che cos'è il trasferimento di file? .....	603
Vantaggi del trasferimento di file per la modernizzazione dei mainframe di AWS .....	603
Come funziona AWS Mainframe Modernization File Transfer .....	604
Installa un agente di trasferimento file .....	605
Fase 1: Accedere a ISPF .....	605
Fase 2: Allocare un set di dati per z/FS .....	605
Passaggio 3: Formattare il set di dati come z/FS .....	606
Fase 4: Definire il filesystem su z/OS .....	606

Fase 5: Montare il file system .....	606
Fase 6: Verificare il montaggio .....	607
Fase 7: Inserire gli OMV .....	607
Passaggio 8: Impostare la variabile di ambiente della directory di installazione dell'agente ..	607
Fase 9: Impostare la variabile di ambiente della directory di lavoro .....	607
Fase 10: Creare la cartella di lavoro .....	607
Fase 11: Copiare il pacchetto tar di modernizzazione del AWS mainframe nella directory di lavoro su z/OS .....	608
Passaggio 12: presupponiamo l'utente root .....	608
Configura autorizzazioni e STC .....	609
Crea un utente IAM con credenziali di accesso a lungo termine .....	610
Crea un ruolo IAM che l'agente deve assumere .....	610
Configurazione dell'agente .....	612
Endpoint per il trasferimento dei dati .....	614
Creare un endpoint per il trasferimento dei dati .....	614
Trasferisci attività .....	616
Crea attività di trasferimento .....	616
Visualizza le attività di trasferimento .....	618
Tutorial: Guida introduttiva a File Transfer .....	619
Panoramica .....	619
Passaggio 1: trasferire il pacchetto tar dei binari dell'agente dalla partizione logica del mainframe AWS .....	620
Fase 2: Configurare l'agente File Transfer sul mainframe di origine .....	620
Fase 3: Creare un endpoint per il trasferimento dei dati .....	620
Passaggio 4: Creare un'attività di trasferimento .....	620
Passaggio 5: Visualizza lo stato di avanzamento dell'operazione di trasferimento .....	620
Sicurezza .....	621
Protezione dei dati .....	622
Dati raccolti da Mainframe AWS Modernization .....	623
Crittografia dei dati inattiva per il servizio di AWS modernizzazione del mainframe .....	624
In che modo AWS Mainframe Modernization utilizza le sovvenzioni in AWS KMS .....	626
Crea una chiave gestita dal cliente .....	628
Specificazione di una chiave gestita dal cliente per la modernizzazione del mainframe AWS .....	630
AWS Modernizzazione del mainframe: contesto di crittografia .....	631
Monitoraggio delle chiavi di crittografia .....	632



Ulteriori informazioni .....	648
Crittografia in transito .....	648
Identity and Access Management .....	648
Destinatari .....	649
Autenticazione con identità .....	650
Gestione dell'accesso con policy .....	653
Come funziona la modernizzazione AWS del mainframe con IAM .....	656
Esempi di policy basate su identità .....	669
Risoluzione dei problemi .....	672
Uso di ruoli collegati ai servizi .....	674
Convalida della conformità .....	677
Resilienza .....	678
Sicurezza dell'infrastruttura .....	679
AWS PrivateLink .....	679
Considerazioni .....	680
Creazione di un endpoint di interfaccia .....	680
Creazione di una policy dell'endpoint .....	680
Monitoraggio .....	682
Monitoraggio con CloudWatch .....	682
Metriche dell'ambiente di runtime .....	683
Parametri di applicazioni .....	684
Dimensioni .....	688
CloudTrail registri .....	688
AWSInformazioni sulla modernizzazione del mainframe in CloudTrail .....	689
Informazioni sulle voci dei file di registro di AWS Mainframe Modernization .....	690
Risoluzione dei problemi .....	692
Errore: timeout in attesa che il nome del set di dati venga sbloccato .....	692
Come si verifica questo errore .....	692
Come fai a sapere se questa è la tua situazione? .....	693
Cosa puoi fare? .....	693
Forza lo sblocco del lucchetto .....	693
Configura il meccanismo di riparazione automatica Blusam .....	694
Blusam locks manager .....	695
Impossibile accedere all'URL di un'applicazione .....	695
Come si verifica questo errore .....	696
Come fai a sapere se questa è la tua situazione? .....	696

---

Cosa puoi fare? .....	696
AWSBlu Insights non si apre dalla console .....	697
Come si verifica questo errore .....	697
Cosa puoi fare? .....	697
Cronologia dei documenti .....	699
.....	dcii

# Cos'è la modernizzazione del AWS mainframe?

AWS La modernizzazione del mainframe consente di modernizzare le applicazioni mainframe in ambienti di runtime gestiti. AWS Offre strumenti e risorse per aiutarti a pianificare e implementare la migrazione e la modernizzazione. È possibile analizzare le applicazioni mainframe esistenti, svilupparle o aggiornarle utilizzando COBOL o PL/I e implementare una pipeline automatizzata per l'integrazione e la distribuzione continue (CI/CD) delle applicazioni. È possibile scegliere tra modelli automatizzati di refactoring e replatforming, a seconda delle esigenze dei clienti. Se sei un consulente che aiuta un cliente a migrare i propri carichi di lavoro mainframe, puoi utilizzare gli strumenti di modernizzazione del AWS mainframe per tutte le fasi del percorso di migrazione e modernizzazione, dalla pianificazione iniziale alle operazioni cloud post-migrazione.

Puoi utilizzare AWS Mainframe Modernization per creare e gestire in modo efficiente l'ambiente di runtime per le tue applicazioni mainframe, nonché AWS per gestire e monitorare le tue applicazioni modernizzate.

## Argomenti

- [Caratteristiche AWS della modernizzazione del mainframe](#)
- [Modelli](#)
- [Come iniziare con la modernizzazione del mainframe AWS](#)
- [Servizi correlati](#)
- [Accesso alla modernizzazione del AWS mainframe](#)
- [Sei un utente di Mainframe Modernization per la prima volta? AWS](#)
- [Prezzi](#)

### Note

Avete collaborato con AWS Mainframe Migration Competency Partners o AWS Professional Services per il vostro progetto di modernizzazione del mainframe? In caso contrario, ti consigliamo vivamente di coinvolgere esperti per il tuo progetto.

- [AWS Partner competenti per la modernizzazione del mainframe](#)
- [AWS Professional Services](#)

Le caratteristiche e i casi d'uso di AWS Mainframe Modernization supportano un approccio di modernizzazione evolutivo, che offre vantaggi a breve termine migliorando l'agilità e numerose opportunità di ottimizzazione e innovazione in un secondo momento. Per ulteriori informazioni, consulta [Approccio alla modernizzazione](#).

## Caratteristiche AWS della modernizzazione del mainframe

AWS Le funzionalità di modernizzazione del mainframe supportano i seguenti casi d'uso:

- **Valutazione:** la funzionalità di valutazione di AWS Mainframe Modernization può aiutarti a valutare, definire e pianificare un progetto di migrazione e modernizzazione.
- **Refactor:** basato su AWS Blu Age, puoi utilizzare il refactoring per convertire i linguaggi di programmazione delle applicazioni legacy, creare macroservizi o microservizi e modernizzare le interfacce utente (UI) e gli stack di software applicativi.

AWS Blu Insights è ora disponibile tramite Single Sign-On. AWS Management Console Non è più necessario gestire credenziali AWS Blu Insights separate. È possibile accedere alle funzionalità di AWS AWS Blu Age Codebase e Transformation Center direttamente da AWS Management Console

- **Replatform:** grazie alla soluzione Micro Focus Enterprise, è possibile trasferire l'applicazione dove gran parte del codice sorgente dell'applicazione viene ricompilato senza modifiche.
- **IDE per sviluppatori:** AWS Mainframe Modernization offre un ambiente di sviluppo integrato (IDE) su richiesta che consente agli sviluppatori di scrivere codice più rapidamente con modifiche e debug intelligenti, compilazione istantanea del codice e test unitari.
- **Runtime gestito:** l'ambiente di esecuzione gestita di AWS Mainframe Modernization monitora continuamente i cluster per mantenere attivi i carichi di lavoro aziendali con elaborazione con riparazione automatica e scalabilità automatizzata.
- **Integrazione e distribuzione continue (CI/CD):** la funzionalità CI/CD di AWS Mainframe Modernization aiuta i team di sviluppo delle applicazioni a apportare modifiche al codice con maggiore frequenza e affidabilità, il che accelera la velocità di migrazione, aumenta la qualità e aiuta a ridurre il rilascio di nuove funzioni aziendali. time-to-market
- **Integrazioni con altri AWS servizi:** AWS Mainframe Modernization supporta AWS CloudFormation e garantisce un'implementazione ripetibile e una maggiore sicurezza e conformità. AWS PrivateLink AWS Key Management Service
- **Disponibilità estesa:** la modernizzazione del AWS mainframe è ora disponibile negli Stati Uniti orientali (Ohio), Stati Uniti occidentali (California settentrionale), Asia Pacifico (Mumbai), Asia

Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Tokyo), Europa (Londra) ed Europa (Parigi).

Per ulteriori informazioni sulle funzionalità di modernizzazione del mainframe, vedere AWS . <https://aws.amazon.com/mainframe-modernization/features/>

## Modelli

Il modello Automated Refactoring, basato su AWS Blu Age, si concentra sull'accelerazione della modernizzazione convertendo l'intero stack di applicazioni legacy e il relativo livello di dati in una moderna applicazione basata su Java, preservando al contempo l'equivalenza funzionale. Durante questa trasformazione automatizzata, crea un'applicazione multilivello con un front-end basato su Angular, un backend Java abilitato all'API e un livello di dati che accede ai moderni archivi di dati. Il processo di refactoring offre funzionalità equivalenti allo stack precedente per aumentare l'automazione dei progetti con conseguente velocità, qualità e costi inferiori per ottenere vantaggi aziendali più rapidamente. Per ulteriori informazioni, vedere [AWS Mainframe Modernization Automated Refactor](#).

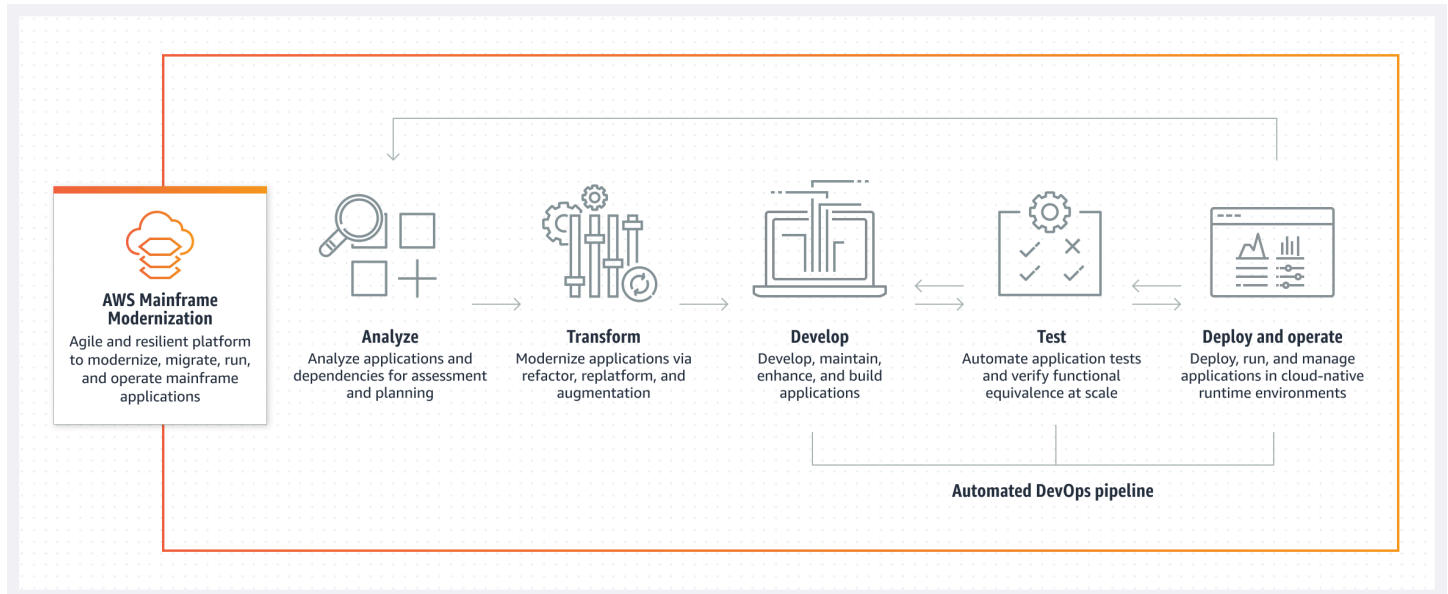
Il modello Replatforming, basato sulla suite Micro Focus Enterprise, si concentra sulla conservazione del linguaggio, del codice e degli artefatti dell'applicazione per ridurre al minimo l'impatto sugli asset e sui team dell'applicazione. Aiuta i clienti a mantenere le conoscenze e le competenze applicative. Sebbene le modifiche alle applicazioni siano limitate, questo modello facilita anche la modernizzazione dell'infrastruttura e dei processi. L'infrastruttura viene trasformata in un moderno servizio gestito basato sul cloud, mentre i processi vengono modificati per seguire le migliori pratiche per lo sviluppo delle applicazioni e le operazioni IT. Per ulteriori informazioni, consulta [AWS Mainframe Modernization Replatform](#).

## Come iniziare con la modernizzazione del mainframe AWS

Eseguire una prova Offriamo tutorial ed esempi di applicazioni per aiutarti a farti un'idea di ciò AWS che offre la modernizzazione del mainframe. Scegli il [Tutorial: Managed Runtime per AWS Blu Age](#) o il [Tutorial: runtime gestito per Micro Focus](#) per un tutorial completo. step-by-step

Se sei interessato al refactoring automatizzato, dai un'occhiata agli strumenti AWS Blu Age all'indirizzo. [BluInsights](#) È inoltre possibile configurare AppStream 2.0 per accedere all'IDE AWS Blu Age Developer o agli strumenti Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer.

I tutorial e le applicazioni di esempio ti danno solo un'idea di ciò che offre la modernizzazione del AWS mainframe. Quando sei pronto per iniziare un progetto di modernizzazione, consulta i dettagli sulle fasi e [Approccio alla modernizzazione](#) le attività di un progetto di modernizzazione.



## Servizi correlati

Oltre a Blu Insights per il refactoring automatizzato, puoi utilizzare i seguenti AWS servizi con Mainframe Modernization. AWS

- Amazon RDS per ospitare i database migrati.
- Amazon S3 per l'archiviazione di file binari e di definizione delle applicazioni.
- Amazon FSx o Amazon EFS per l'archiviazione dei dati delle applicazioni.
- Amazon AppStream per l'accesso agli strumenti Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer.
- AWS CloudFormation per la DevOps pipeline automatizzata che puoi utilizzare per configurare CI/CD per le tue applicazioni migrate.
- AWS Migration Hub
- AWS DMS per la migrazione dei database.

## Accesso alla modernizzazione del AWS mainframe

[Attualmente, è possibile accedere a AWS Mainframe Modernization tramite la console all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/). Per un elenco delle regioni in cui è disponibile la modernizzazione del AWS mainframe, consulta Endpoint e quote per la [modernizzazione AWS del mainframe](#) nel. Riferimenti generali di Amazon Web Services

## Sei un utente di Mainframe Modernization per la prima volta? AWS

Se utilizzi per la prima volta AWS Mainframe Modernization, ti consigliamo di iniziare leggendo le seguenti sezioni:

- [Nozioni di base](#)
- [Configurazione](#)

## Prezzi

AWS La modernizzazione del mainframe addebita i costi per l'utilizzo di istanze che supportano gli ambienti di runtime gestiti. Inoltre, AWS Mainframe Modernization offre alcuni strumenti senza costi aggiuntivi. L'utente è responsabile del pagamento delle tariffe sostenute per altri AWS servizi utilizzati in relazione alla modernizzazione del mainframe. AWS AWS fornirà un preavviso di 30 giorni prima che qualsiasi modifica dei prezzi entri in vigore per l'uso di Mainframe Modernization. AWS Per ulteriori informazioni, consulta [Mainframe](#) Modernization with. AWS

Con AWS Blu Insights, paghi per l'utilizzo di Transformation Center. Per ulteriori informazioni, consulta i prezzi di [AWS Mainframe Modernization](#).

# Configurazione della modernizzazione del AWS mainframe

Prima di iniziare a utilizzare AWS Mainframe Modernization, tu o il tuo amministratore dovete completare alcuni passaggi.

Argomenti

- [Registrati per un Account AWS](#)
- [Creazione di un utente amministratore](#)

## 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 AWS viene 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 Console](#) come 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 [Accesso come utente root](#) 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 Centro identità AWS IAM, assegna l'accesso amministrativo a un utente amministrativo.

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.

# Guida introduttiva alla modernizzazione del AWS mainframe

Per iniziare a usare AWS Mainframe Modernization, puoi seguire dei tutorial che ti introducono al servizio e a ciascun motore di runtime.

## Argomenti

- [Tutorial: Managed Runtime per AWS Blu Age](#)
- [Tutorial: runtime gestito per Micro Focus](#)

Per continuare a imparare, consulta i seguenti tutorial.

- [Tutorial: configurazione della build Micro Focus per l'applicazione BankDemo di esempio](#)
- [Tutorial: configurazione di una pipeline CI/CD da utilizzare con Micro Focus Enterprise Developer](#)

## Tutorial: Managed Runtime per AWS Blu Age

Questo tutorial mostra come implementare un'applicazione modernizzata AWS Blu Age in un ambiente di runtime di modernizzazione del AWS mainframe.

## Argomenti

- [Prerequisiti](#)
- [Passaggio 1: carica l'applicazione demo](#)
- [Fase 2: Creare la definizione dell'applicazione](#)
- [Fase 3: Creare un ambiente di runtime](#)
- [Fase 4: Creare un'applicazione](#)
- [Fase 5: Implementazione di un'applicazione](#)
- [Fase 6: Avviare un'applicazione](#)
- [Passaggio 7: Accedere all'applicazione](#)
- [Fase 8: test dell'applicazione](#)
- [Pulizia delle risorse](#)

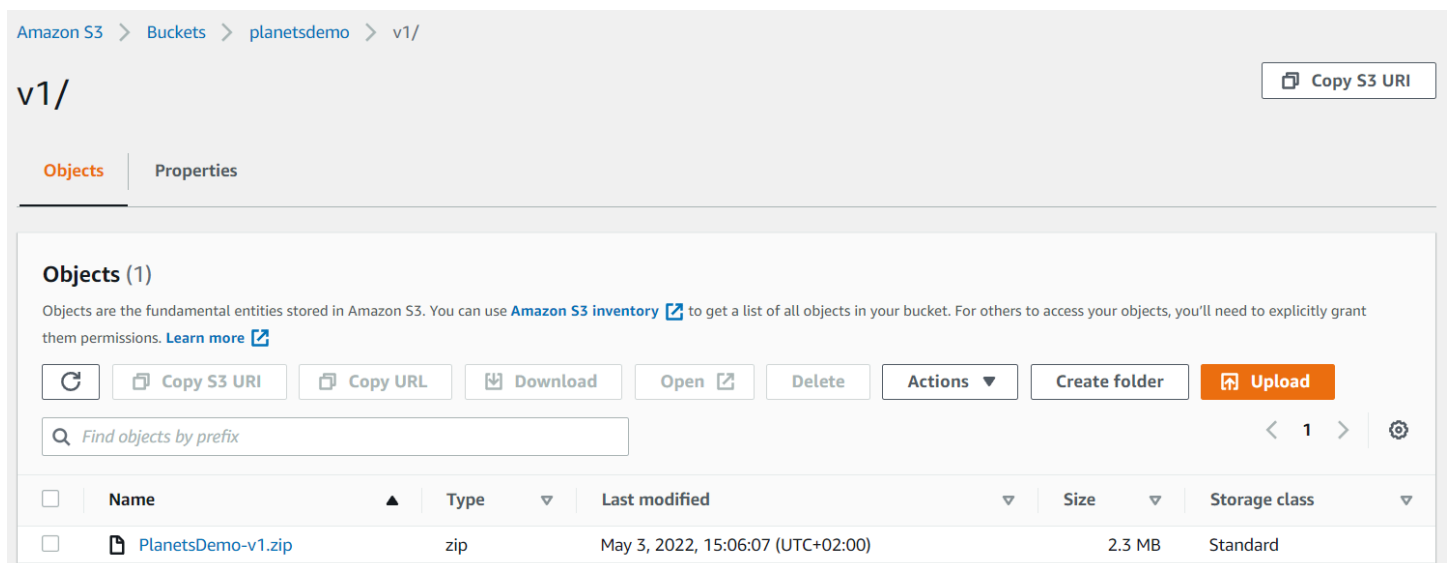
## Prerequisiti

[Per completare questo tutorial, scaricate l'archivio dell'applicazione demo -v1.zip. PlanetsDemo](#)

L'applicazione demo in esecuzione richiede un browser moderno per l'accesso. Se esegui questo browser dal desktop o da un'istanza Amazon Elastic Compute Cloud, ad esempio all'interno del VPC, determina le tue impostazioni di sicurezza.

## Passaggio 1: carica l'applicazione demo

Carica l'applicazione demo in un bucket Amazon S3. Assicurati che questo bucket si trovi nello stesso Regione AWS punto in cui distribuirai l'applicazione. L'esempio seguente mostra un bucket denominato planetsdemo, con un prefisso chiave, o cartella, denominato v1 e un archivio denominato. planetsdemo-v1.zip



The screenshot shows the Amazon S3 console interface. The breadcrumb navigation is 'Amazon S3 > Buckets > planetsdemo > v1/'. The current view is 'v1/' with a 'Copy S3 URI' button. Below the navigation are tabs for 'Objects' and 'Properties'. The 'Objects' tab is active, showing a list of objects. The list has one entry: 'PlanetsDemo-v1.zip' with a type of 'zip', last modified on 'May 3, 2022, 15:06:07 (UTC+02:00)', a size of '2.3 MB', and a storage class of 'Standard'. Above the list are various action buttons: Refresh, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. A search bar is also present with the placeholder 'Find objects by prefix'.

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	PlanetsDemo-v1.zip	zip	May 3, 2022, 15:06:07 (UTC+02:00)	2.3 MB	Standard

### Note

La cartella nel bucket è obbligatoria.

## Fase 2: Creare la definizione dell'applicazione

Per distribuire un'applicazione nel runtime gestito, è necessaria una definizione dell'applicazione di modernizzazione AWS del mainframe. Questa definizione è un file JSON che descrive la posizione e le impostazioni dell'applicazione. L'esempio seguente è una definizione di applicazione di questo tipo per l'applicazione demo:

```
{
  "template-version": "2.0",
  "source-locations": [{
    "source-id": "s3-source",
    "source-type": "s3",
    "properties": {
      "s3-bucket": "planetsdemo",
      "s3-key-prefix": "v1"
    }
  }],
  "definition": {
    "listeners": [{
      "port": 8196,
      "type": "http"
    }],
    "ba-application": {
      "app-location": "${s3-source}/PlanetsDemo-v1.zip"
    }
  }
}
```

Modificate la `s3-bucket` voce con il nome del bucket in cui avete memorizzato il file zip dell'applicazione di esempio.

Per ulteriori informazioni sulla definizione dell'applicazione, vedere [AWS Esempio di definizione dell'applicazione Blu Age](#).

### Fase 3: Creare un ambiente di runtime

Per creare l'ambiente di runtime di modernizzazione del AWS mainframe, effettuate le seguenti operazioni:

1. Apri la console di modernizzazione del [AWSmainframe](#).
2. Nel Regione AWS selettore, scegli la regione in cui desideri creare l'ambiente. Regione AWSDeve corrispondere alla regione in cui hai creato il bucket S3. [Passaggio 1: carica l'applicazione demo](#)
3. In Modernizza le applicazioni mainframe, scegli Refactor with Blu Age, quindi scegli Inizia.

## Modernize mainframe applications

Analyze your applications, make changes to them, and deploy them on a runtime environment.

Choose an option to get started.






- Refactor with Blu Age
- Replatform with Micro Focus

**Get started**

4. In Come può aiutare AWS Mainframe Modernization, scegli Deploy and Create runtime environment.

### How can AWS Mainframe Modernization help?

AWS Mainframe Modernization supports migration, modernization, and optimization; maintenance and incremental improvements; and ongoing operation and execution.

<input type="radio"/> Analyze/Refactor 	<input type="radio"/> Develop 	<input checked="" type="radio"/> Deploy 
<input type="radio"/> Test 	<b>Operate</b> <a href="#">Info</a> 	

#### Deploy [Info](#)

- Create runtime environment**  
Create a runtime environment with Blu Age engine for applications.
- Create application**  
Create applications and deploy them in the runtime environment.

5. Nella barra di navigazione a sinistra, scegli Ambienti, quindi scegli Crea ambiente. Nella pagina Specificare le informazioni di base, inserisci un nome e una descrizione per il tuo ambiente,

quindi assicurati che il motore AWS Blu Age sia selezionato. Facoltativamente, puoi aggiungere tag alla risorsa creata. Quindi scegli Successivo.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
**Specify basic information**

Step 2  
Specify configurations

Step 3 - *Optional*  
Attach storage

Step 4  
Review and create

## Specify basic information [Info](#)

### Name and description

Environment name


Use only alphanumeric characters, hyphens, and underscores. The maximum length is 60 characters.


Environment description - *optional*

The description can be up to 500 characters.

### Engine options

Select Engine

**AWS Blu Age**  
This engine provides the framework and dependencies necessary to execute applications refactored by Blu Age.  


**Micro Focus**  
The engine provides a mainframe-compatible runtime for replatformed applications by Micro Focus.  


AWS Blu Age Version

6. Nella pagina Specificare le configurazioni, scegli Ambiente di runtime autonomo.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
**Specify configurations**

Step 3 - Optional  
Attach storage

Step 4  
Review and create

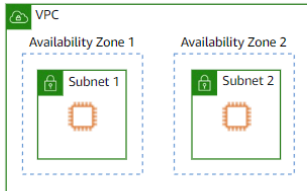
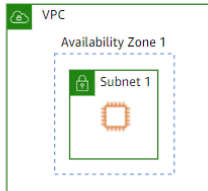
## Specify configurations Info

### Availability Info

Choose the availability pattern for your environment.

**Standalone runtime environment**  
Sets up a single instance in a single availability zone. Does not guarantee high availability but costs less.

**High availability cluster**  
Sets up redundant instances across two availability zones. Enables higher availability but costs more.



## 7. In Sicurezza e rete, apporta le seguenti modifiche:

- Scegli Consenti l'accesso pubblico alle applicazioni distribuite in questo ambiente. Questa opzione assegna un indirizzo IP pubblico all'applicazione in modo che sia possibile accedervi dal desktop.
- Selezione di un VPC. È possibile utilizzare l'impostazione predefinita.
- Scegli due sottoreti. Assicurati che le sottoreti consentano l'assegnazione di indirizzi IP pubblici.
- Scelta del gruppo di sicurezza. È possibile utilizzare l'impostazione predefinita. Assicuratevi che il gruppo di sicurezza scelto consenta l'accesso dall'indirizzo IP del browser alla porta specificata nella `listener` proprietà della definizione dell'applicazione. Per ulteriori informazioni, consulta [Fase 2: Creare la definizione dell'applicazione](#).

## Security and network

Allow applications deployed to this environment to be publicly accessible.

**Virtual Private Cloud (VPC)**  
Choose the VPC where you want to create the environment.

Default vpc-

**Subnets**  
Choose one or more subnets for a high availability setup.

Choose subnets

subnet- X

subnet- X

**Security groups**  
Choose one or more security groups for the chosen VPC.

Choose security groups

default X  
default VPC security group

Se desideri accedere all'applicazione dall'esterno del VPC che hai scelto, assicurati che le regole in entrata per quel VPC siano configurate correttamente. Per ulteriori informazioni, consulta la pagina [Impossibile accedere all'URL di un'applicazione](#).

8. Seleziona Next (Successivo).
9. In Allega archiviazione - Facoltativo, lascia le selezioni predefinite e scegli Avanti.



AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
Specify configurations

Step 3 - *Optional*  
**Attach storage**

Step 4  
Review and create

## Attach storage - *Optional* Info

### EFS storage

Choose one or more existing EFS file systems. Specify a mount point for each system.

No EFS associated with this environment.

**Choose EFS storage**

You can add up to 1 more EFS.

### FSx storage

Choose one or more existing FSx for Lustre file systems. Specify a mount point for each system.

No EFS associated with this environment.

**Choose FSx storage**

You can add up to 1 more FSx.

Cancel Previous **Next**

10. In Pianifica manutenzione, scegli Nessuna preferenza, quindi scegli Avanti.

11. In Rivedi e crea, esamina le informazioni, quindi scegli Crea ambiente.

## Fase 4: Creare un'applicazione

1. Passa a AWS Mainframe Modernization in. AWS Management Console
2. Nel riquadro di navigazione, scegli Applications (Applicazioni), quindi Create application (Crea applicazione). Nella pagina Specificare le informazioni di base, inserisci un nome e una descrizione per l'applicazione e assicurati che sia selezionato il motore AWS Blu Age. Quindi scegli Successivo.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name


Use only alphanumeric characters, hyphens, and underscores. The maximum length is 60 characters.

Application description - *optional*


The maximum length is 500 characters.

### Engine type

**AWS Blu Age**  
This engine provides the framework and dependencies necessary to execute applications refactored by Blu Age.



**Micro Focus**  
This engine provides a mainframe-compatible runtime for replatformed applications by Micro Focus



3. Nella pagina Specificare risorse e configurazioni, copia e incolla la definizione JSON dell'applicazione aggiornata in cui è stata creata. [the section called "Fase 2: Creare la definizione dell'applicazione"](#)

AWS Mainframe Modernization > Applications > Create application

Step 1  
Specify basic information

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

## Specify resources and configurations [Info](#)

### Resources and configurations

Choose an approach to define the application

Specify the application definition with its resources and configurations using the inline editor

Use an application definition JSON file in an Amazon S3 bucket

```

1  {
2  "resources": [
3    {
4      "resource-type": "listener",
5      "resource-id": "tomcat",
6      "properties": {
7        "port": 8196,
8        "type": "http"
9      }
10   },
11   {
12     "resource-type": "ba-application",
13     "resource-id": "planetsdemo",
14     "properties": {
15       "app-location": "${s3-source}/PlanetsDemo-v1.zip"
16     }
17   }
18 ],
19 "source-locations": [

```

JSON Ln 29, Col 2 ⊗ Errors: 0 ⚠ Warnings: 0

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

4. In Rivedi e crea, esamina le tue scelte, quindi scegli Crea applicazione.

## Fase 5: Implementazione di un'applicazione

Dopo aver creato correttamente l'ambiente di runtime e l'applicazione AWS Mainframe Modernization ed entrambi nello stato Disponibile, è possibile distribuire l'applicazione nell'ambiente di runtime. Per farlo, completa le seguenti fasi.

1. Passa alla modernizzazione di AWS Mainframe nella console di AWS gestione. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo). Viene visualizzata la pagina con l'elenco degli ambienti.

AWS Mainframe Modernization > Environments

**Environments (1)** [Info](#) ↻ Actions ▾ Create environment

< 1 > ⚙️

<input type="checkbox"/>	Environment name ▾	Status ▾	Engine ▾	Version ▾	Instance type ▾	Creatio... ▾
<input type="checkbox"/>	planets-demo-env	Available	AWS Blu Age	3.1.0	M2.m5.large	May 20, 20...

2. Scegliete l'ambiente di runtime creato in precedenza. Viene visualizzata la pagina dei dettagli dell'ambiente.

3. Scegli Deploy application.

AWS Mainframe Modernization > Environments > planets-demo-env

**planets-demo-env** [Info](#) Actions ▾ Deploy application

[Summary](#) | [Configurations](#) | [Deployed applications](#) | [Tags](#)

**Environment** [Info](#)

Name planets-demo-env	Description -	Engine AWS Blu Age 3.1.0	Availability Standalone
ARN arn:aws:m2:	Deployed applications 0	Status Available	Creation time May 20, 2022, 10:46 (UTC+02:00)

**Applications summary** [Info](#)

**No applications**  
No applications to display.  
Deploy application

4. Scegli l'applicazione creata in precedenza, quindi scegli la versione in cui vuoi distribuire l'applicazione. Scegliere quindi Deploy (Distribuisci).

[AWS Mainframe Modernization](#) > [Applications](#) > [my-ba-planetsdemo](#) > **Deploy application**

## Deploy application Info

You have selected the following application:

Name	Description	Engine
my-ba-planetsdemo	Runtime environment for the PlanetsDemo App.	Blu Age

### Available versions (0) ↻

Choose a version from the list.

< 1 > ⚙️

Version

Creation time

No versions  
No versions to display

### Environments (1) Info

< 1 > ⚙️

Enviro...	Status	Engine	Version	Instance type	Cr
<input type="radio"/> <a href="#">planets-demo-e</a>	<span>✔️ Available</span>	Blu Age	3.7.0	M2.m5.large	De

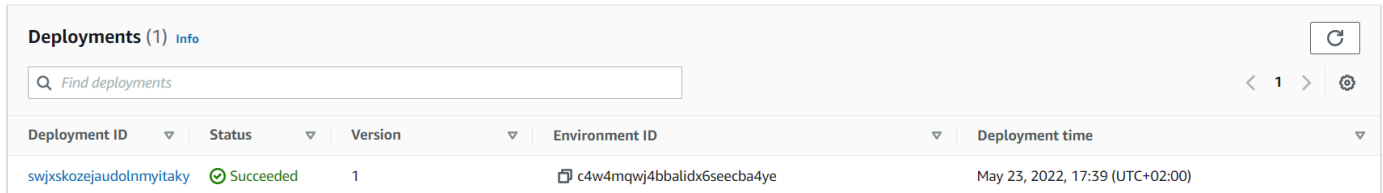
Cancel

Deploy

5. Attendi che l'applicazione finisca la distribuzione. Verrà visualizzato un banner con il messaggio L'applicazione è stata distribuita correttamente.

## Fase 6: Avviare un'applicazione

1. Passa a AWS Mainframe Modernization in AWS Management Console e scegli Applicazioni.
2. Scegli la tua applicazione, quindi vai a Distribuzioni. Lo stato della domanda dovrebbe essere Riuscito.



The screenshot shows the 'Deployments (1)' section in the AWS Management Console. It features a search bar with the placeholder 'Find deployments', a refresh button, and a table with the following columns: Deployment ID, Status, Version, Environment ID, and Deployment time. A single deployment is listed with a 'Succeeded' status.

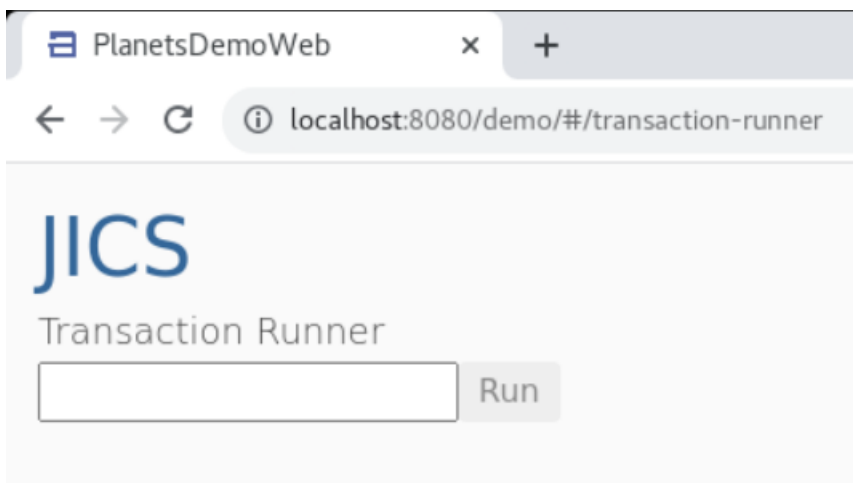
Deployment ID	Status	Version	Environment ID	Deployment time
swjxskozejaudolnmyitaky	Succeeded	1	c4w4mqwj4bbalidx6seecba4ye	May 23, 2022, 17:39 (UTC+02:00)

3. Scegli Azioni, quindi scegli Avvia applicazione.

## Passaggio 7: Accedere all'applicazione

1. Attendi che l'applicazione sia nello stato In esecuzione. Verrà visualizzato un banner con il messaggio L'applicazione è stata avviata correttamente.
  2. Copia il nome host DNS dell'applicazione. È possibile trovare questo nome host nella sezione Informazioni sull'applicazione dell'applicazione.
  3. In un browser, vai a `http://{hostname}:{portname}/PlanetsDemo-web-1.0.0/`, dove:
    - `hostname` è il nome host DNS copiato in precedenza.
    - `portname` è la porta Tomcat definita nella definizione dell'applicazione in cui è stata creata.
- [Fase 2: Creare la definizione dell'applicazione](#)

Viene visualizzata la schermata JICS.



Se non riesci ad accedere all'applicazione, consulta. [Impossibile accedere all'URL di un'applicazione](#)

### Note

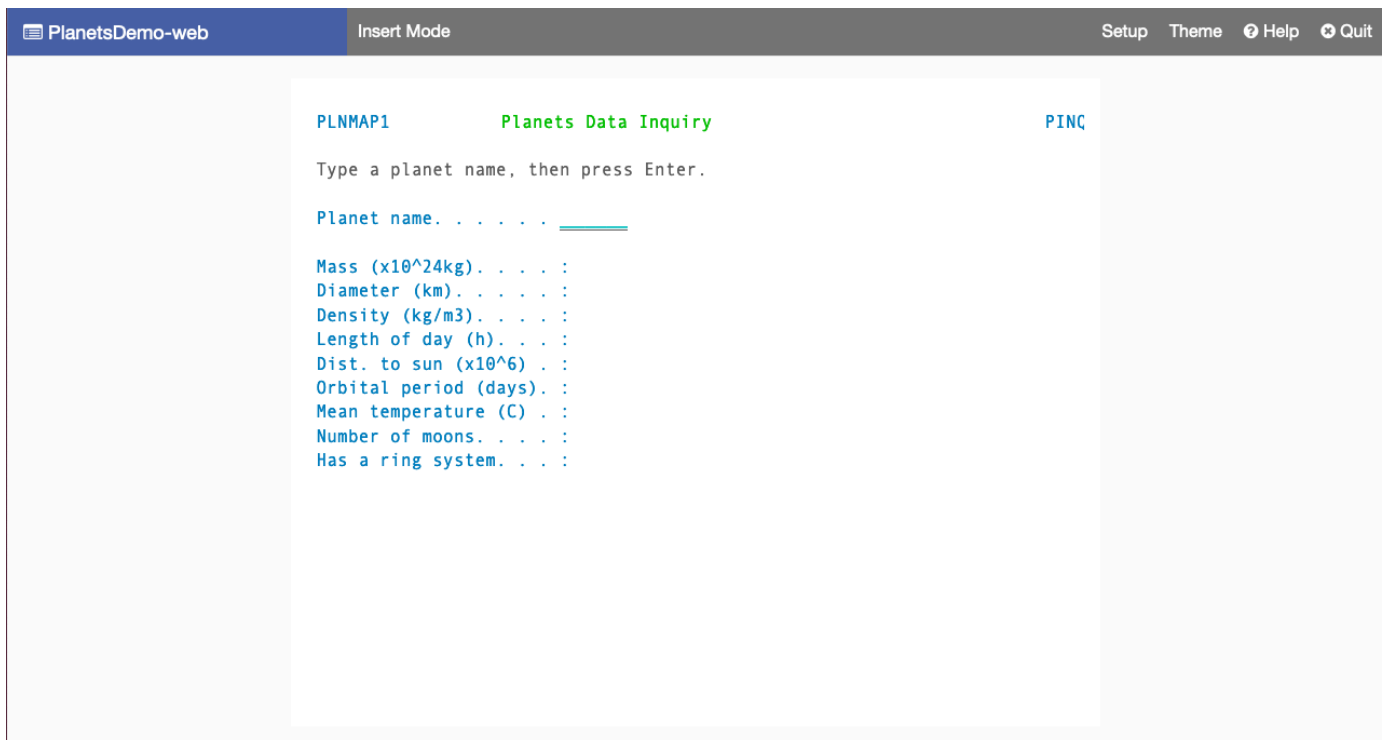
Se l'applicazione non è accessibile e la regola in entrata sul gruppo di sicurezza ha selezionato «My IP» sulla porta 8196, specifica la regola per consentire il traffico da LB i/p sulla porta 8196.

## Fase 8: test dell'applicazione

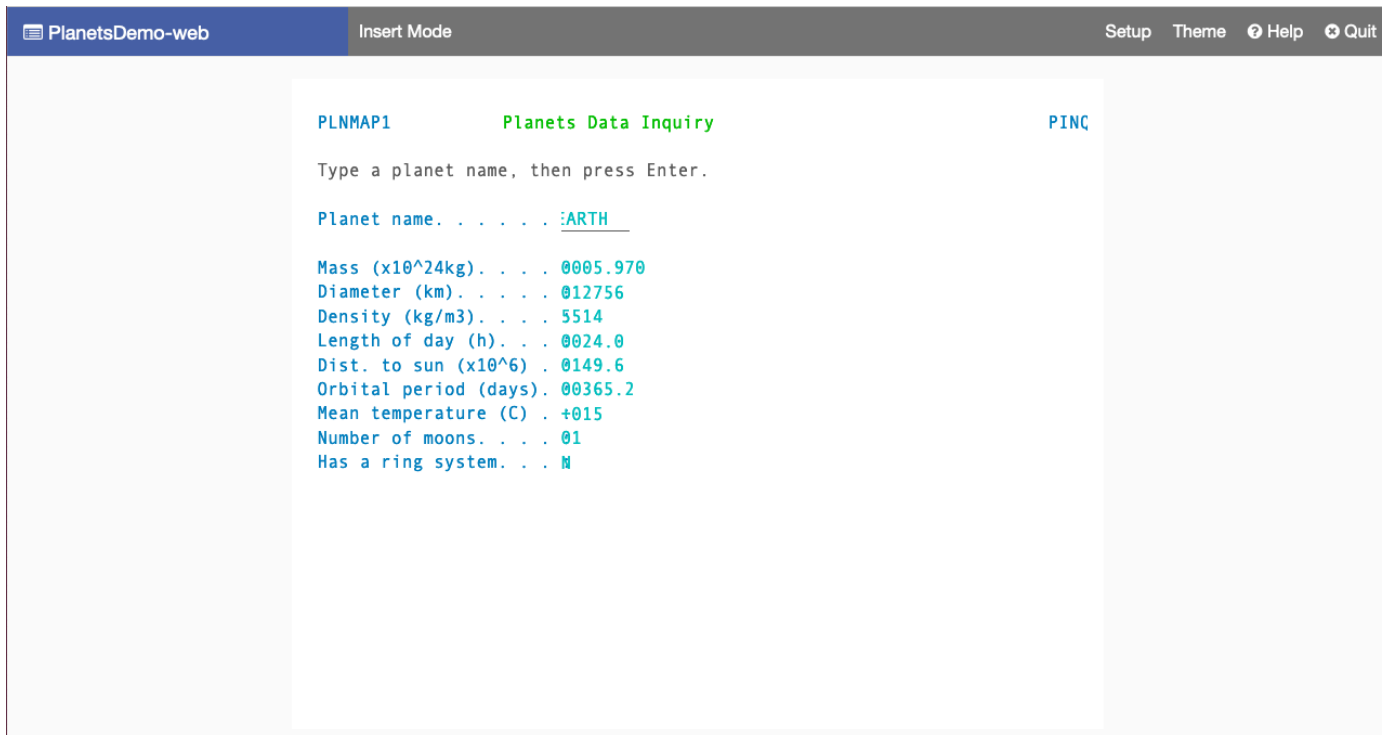
In questo passaggio, si esegue una transazione nell'applicazione migrata.

1. Nella schermata JICS, inserisci PINQ il campo di immissione e scegli Esegui (o premi Invio) per avviare la transazione dell'applicazione.

Dovrebbe apparire la schermata dell'app demo.



2. Digita il nome di un pianeta nel campo corrispondente e premi Invio.



```
PlanetsDemo-web Insert Mode Setup Theme Help Quit
PLNMAP1 Planets Data Inquiry PINC
Type a planet name, then press Enter.
Planet name. . . . . :ARTH
Mass (x10^24kg). . . . 0005.970
Diameter (km). . . . . 012756
Density (kg/m3). . . . 5514
Length of day (h). . . 0024.0
Dist. to sun (x10^6) . 0149.6
Orbital period (days). 00365.2
Mean temperature (C) . +015
Number of moons. . . . 01
Has a ring system. . . N
```

Dovresti vedere i dettagli sul pianeta.

## Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale per evitare costi aggiuntivi. Per fare ciò, completa la seguente procedura:

- Se l'applicazione AWS Mainframe Modernization è ancora in esecuzione, interrompila.
- Elimina l'applicazione . Per ulteriori informazioni, consulta [Eliminare un'applicazione di modernizzazione AWS del mainframe](#).
- Eliminare l'ambiente di runtime. Per ulteriori informazioni, consulta [Eliminare un ambiente di runtime di AWS modernizzazione del mainframe](#).

## Tutorial: runtime gestito per Micro Focus

Questo tutorial mostra come distribuire ed eseguire l'applicazione di CardDemo esempio in un ambiente di runtime gestito da AWS Mainframe Modernization con il motore di runtime Micro Focus. L'applicazione CardDemo di esempio è un'applicazione semplificata per carte di credito sviluppata



per testare e presentare ai partner la tecnologia per i casi d'uso della AWS modernizzazione del mainframe.

Nel tutorial, crei risorse in altro. Servizi AWS Questi includono Amazon Simple Storage Service, Amazon Relational Database Service AWS Key Management Service e. AWS Secrets Manager

## Argomenti

- [Prerequisiti](#)
- [Fase 1: creare e caricare un bucket Amazon S3](#)
- [Passaggio 2: creare e configurare un database](#)
- [Passaggio 3: creare e configurare un AWS KMS key](#)
- [Passaggio 4: Creare e configurare un AWS Secrets Manager database segreto](#)
- [Fase 5: Creare un ambiente di runtime](#)
- [Fase 6: Creare un'applicazione](#)
- [Fase 7: Implementazione di un'applicazione](#)
- [Fase 8: Importazione dei set di dati](#)
- [Fase 9: Avviare un'applicazione](#)
- [Fase 10: Connect all'applicazione CardDemo CICS](#)
- [Pulizia delle risorse](#)
- [Passaggi successivi](#)

## Prerequisiti

- Assicurati di avere accesso a un emulatore 3270 per utilizzare la connessione CICS. Gli emulatori 3270 gratuiti e di prova sono disponibili su siti Web di terze parti. In alternativa, è possibile avviare un'istanza Micro Focus di AWS Mainframe Modernization AppStream 2.0 e utilizzare l'emulatore Rumba 3270 (non disponibile gratuitamente).

Per informazioni sulla versione 2.0, vedere. AppStream [the section called “Tutorial: Configurazione AppStream 2.0 per Enterprise Analyzer ed Enterprise Developer”](#)

**Note**

Durante la creazione dello stack, scegliete l'opzione Enterprise Developer (ED) e non Enterprise Analyzer (EA).

- Scaricate l'[applicazione CardDemo di esempio](#) e decomprimate il file scaricato in qualsiasi directory locale. Questa directory conterrà una sottodirectory intitolata. CardDemo
- Identifica un VPC nel tuo account dove puoi definire le risorse create in questo tutorial. Il VPC avrà bisogno di sottoreti in almeno due zone di disponibilità. Per ulteriori informazioni su Amazon VPC, consulta Come funziona [Amazon VPC](#).

## Fase 1: creare e caricare un bucket Amazon S3

In questo passaggio, crei un bucket Amazon S3 e carichi CardDemo i file in questo bucket. Più avanti in questo tutorial, utilizzerai questi file per distribuire ed eseguire l'applicazione di CardDemo esempio in un ambiente Micro Focus Managed Runtime di modernizzazione AWS del mainframe.

**Note**

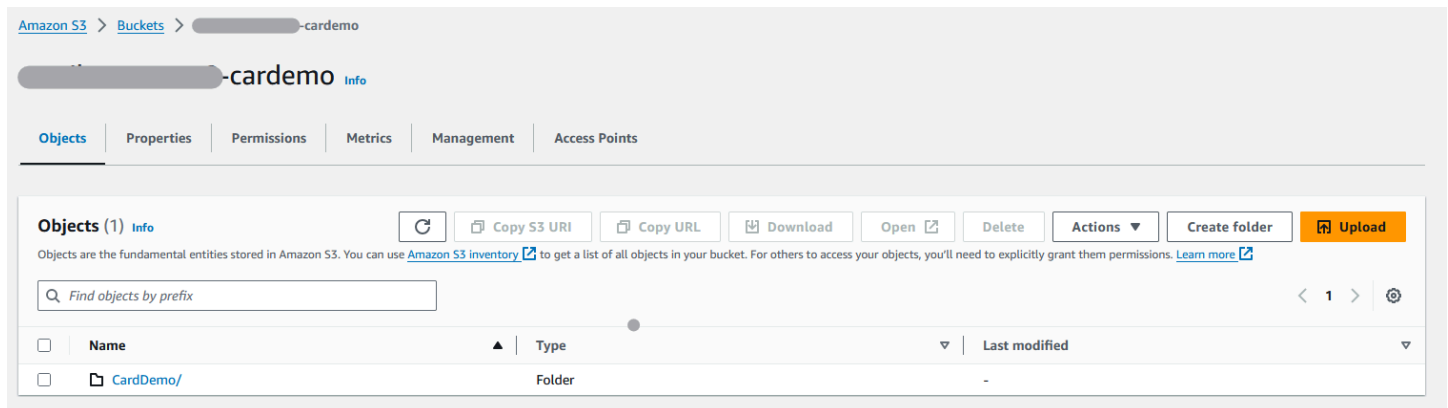
Non è necessario creare un nuovo bucket S3, ma il bucket scelto deve trovarsi nella stessa regione delle altre risorse utilizzate in questo tutorial.

Come creare un bucket Amazon S3.

1. Apri la [console Amazon S3](#) e scegli Crea bucket.
2. Nella configurazione generale, scegli la regione AWS in cui desideri creare il AWS Mainframe Modernization Micro Focus Managed Runtime.
3. Inserisci il nome di un Bucket, ad esempio. `yourname-aws-region-carddemo` Mantieni le impostazioni predefinite e scegli Crea bucket. In alternativa, puoi anche copiare le impostazioni da un bucket Amazon S3 esistente e quindi scegliere Crea bucket.
4. Scegli il bucket che hai appena creato, quindi scegli Carica.
5. Nella sezione Carica, scegli Aggiungi cartella, quindi accedi alla CardDemo directory dal tuo computer locale.

6. Scegli Carica per avviare il processo di caricamento. I tempi di caricamento variano in base alla velocità di connessione.
7. Al termine del caricamento, conferma che tutti i file siano stati caricati correttamente, quindi scegli Chiudi.

Il tuo bucket Amazon S3 ora contiene la cartella. CardDemo



Per informazioni sui bucket S3, consulta [Creazione, configurazione e utilizzo dei bucket Amazon S3](#).

## Passaggio 2: creare e configurare un database

In questo passaggio, crei un database PostgreSQL in Amazon Relational Database Service (Amazon RDS). Per il tutorial, questo database contiene i set di dati utilizzati dall'applicazione di CardDemo esempio per le attività dei clienti relative alle transazioni con carta di credito.

Per creare un database in Amazon RDS

1. Apri la [console Amazon RDS](#).
2. Scegli la regione AWS in cui desideri creare l'istanza di database.
3. Nel pannello di navigazione seleziona Databases (Database).
4. Scegli Crea database, quindi scegli Standard create.
5. Per Tipo di motore scegli PostgreSQL.
6. Scegli una versione del motore 15 o superiore.

### Note

Salva la versione del motore perché ne avrai bisogno più avanti in questo tutorial.

7. Nella sezione Modelli, seleziona Piano gratuito.
8. Cambia l'identificatore dell'istanza DB con qualcosa di significativo, ad esempio. `MicroFocus-Tutorial`
9. Evita di gestire le credenziali principali in. AWS Secrets Manager Inserisci invece una password principale e confermala.


 Note

Salva il nome utente e la password che usi per il database. Li archiverai in modo sicuro nei passaggi successivi di questo tutorial.

10. In Connettività, scegli il VPC in cui desideri creare l'ambiente di runtime gestito per la modernizzazione del AWS mainframe.
11. Scegliere Crea database.

Per creare un gruppo di parametri personalizzato in Amazon RDS

1. Nel riquadro di navigazione della console Amazon RDS, scegli Gruppi di parametri, quindi scegli Crea gruppo di parametri.
2. Nella finestra Crea gruppo di parametri, per la famiglia di gruppi di parametri, seleziona l'opzione Postgres che corrisponde alla versione del tuo database.

 Note

Alcune versioni di Postgres richiedono un Type. Se necessario, selezionate DB Parameter Group. Inserite un nome di gruppo e una descrizione per il gruppo di parametri.


3. Scegli Crea.

Per configurare il gruppo di parametri personalizzato

1. Scegliete il gruppo di parametri appena creato.
2. Seleziona Actions (Operazioni), quindi scegli Edit (Modifica).
3. Filtra `max_prepared_transactions` e modifica il valore del parametro su 100.
4. Seleziona Salva modifiche.

Per associare il gruppo di parametri personalizzato al database


1. Nel riquadro di navigazione della console Amazon RDS, scegli Databases, quindi scegli l'istanza di database che desideri modificare.
2. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).

 Note

L'opzione Modifica non è disponibile fino al termine della creazione e del backup del database, operazione che potrebbe richiedere alcuni minuti.

3. Nella pagina Modifica istanza DB, accedi a Configurazione aggiuntiva e modifica il gruppo di parametri DB con il tuo gruppo di parametri. Se il gruppo di parametri non è disponibile nell'elenco, controlla se è stato creato con la versione corretta del database.
4. Scegli Continua e controlla il riepilogo delle modifiche.
5. Scegli Applica immediatamente per applicare le modifiche all'istante.
6. Scegliere Modifica istanza database per salvare le modifiche.

Per ulteriori informazioni sui gruppi di parametri, consulta [Lavorare con i gruppi di parametri](#).

 Note

Puoi anche usare un database Amazon Aurora PostgreSQL con AWS Mainframe Modernization, ma non esiste un'opzione di livello gratuito. Per ulteriori informazioni, consulta [Lavorare con Amazon Aurora PostgreSQL](#).

## Passaggio 3: creare e configurare un AWS KMS key

Per archiviare le credenziali in modo sicuro per l'istanza Amazon RDS, crea innanzitutto un. AWS KMS key

Per creare un AWS KMS key

1. Apri la console del servizio di [gestione delle chiavi](#).
2. Scegli Create Key (Crea chiave).

3. Lascia le impostazioni predefinite di Symmetric per il tipo di chiave e Encrypt and decrypt per l'utilizzo delle chiavi.
4. Seleziona Avanti.
5. Assegna alla chiave un alias come e una descrizione opzionale. `MicroFocus-Tutorial-RDS-Key`
6. Seleziona Avanti.
7. Assegna un amministratore chiave selezionando la casella accanto all'utente o al ruolo.
8. Scegli Avanti, quindi scegli nuovamente Avanti.
9. Nella schermata di revisione, modifica la politica chiave, quindi inserisci quanto segue:

```
{  
  "Sid" : "Allow access for Mainframe Modernization Service",  
  "Effect" : "Allow",  
  "Principal" : {  
    "Service" : "m2.amazonaws.com"  
  },  
  "Action" : "kms:Decrypt",  
  "Resource" : "*" ,  
},
```

Questa politica concede le autorizzazioni di decrittografia di AWS Mainframe Modernization utilizzando questa politica chiave specifica.

10. Scegli Fine per creare la chiave.

Per ulteriori informazioni, consulta [Creazione di chiavi](#) nella Guida per AWS Key Management Service gli sviluppatori.

## Passaggio 4: Creare e configurare un AWS Secrets Manager database segreto

Ora archivia le credenziali del database in modo sicuro utilizzando and. AWS Secrets Manager AWS KMS key

Per creare e configurare un database segreto AWS Secrets Manager

1. Apri la [console Secrets Manager](#).
2. Nel riquadro di navigazione, scegli Secrets (Segreti).

3. In Segreti, scegli Memorizza un nuovo segreto.
4. Imposta il tipo segreto su Credentials for Amazon RDS database.
5. Inserisci le credenziali che hai specificato quando hai creato il database.
6. In Chiave di crittografia, seleziona la chiave creata nel passaggio 3.
7. Nella sezione Database, seleziona il database creato per questo tutorial, quindi scegli Avanti.
8. In Nome segreto, inserisci un nome come `MicroFocus-Tutorial-RDS-Secret` e una descrizione opzionale.
9. Nella sezione Autorizzazioni per le risorse, scegli Modifica autorizzazioni e sostituisci il contenuto con la seguente politica:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "m2.amazonaws.com"
      },
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "*"
    }
  ]
}
```

10. Selezionare Salva.
11. Scegli Avanti per le schermate successive, quindi scegli Store. Aggiorna l'elenco dei segreti per visualizzare il nuovo segreto.
12. Scegli il segreto appena creato e prendine nota `Secret ARN` perché ti servirà più avanti nel tutorial.
13. Nella scheda Panoramica del segreto, scegli Recupera valore segreto.
14. Scegli Modifica, quindi scegli Aggiungi riga.
15. Aggiungi una chiave per `sslMode` con un valore `verify-full`:

#### Edit secret value

Key/value

Plaintext

sslMode

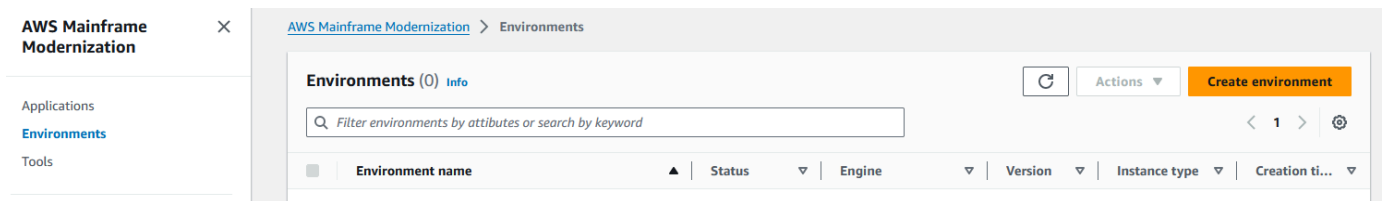
verify-full

## 16. Selezionare Salva.

# Fase 5: Creare un ambiente di runtime

Per creare un ambiente di runtime

1. Apri la console di [modernizzazione AWS del mainframe](#).
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo). Quindi scegli Crea ambiente.



3. In Specificare le informazioni di base,
  - a. Immettete MicroFocus-Environment il nome dell'ambiente.
  - b. Nelle opzioni del motore, assicuratevi che Micro Focus sia selezionato.
  - c. Scegliete la versione più recente di Micro Focus.
  - d. Seleziona Avanti.



## Name and description [Info](#)

### Environment name

Use only alphanumeric characters, hyphens, and underscores. The maximum length is 60 characters.

### Environment description - *optional*

The description can be up to 500 characters.

## Engine options [Info](#)

### Select Engine

Blu Age

This engine provides the framework and dependencies necessary to execute applications refactored by Blu Age.



Micro Focus

The engine provides a mainframe-compatible runtime for replatformed applications by Micro Focus.



### Micro Focus Version

## 4. Configura l'ambiente

- a. In **Disponibilità**, scegli **Cluster ad alta disponibilità**.
- b. In **Risorse**, scegli una **M2.c5.large** o **M2.m5.large** per il tipo di istanza e il numero di istanze che desideri. Specificate fino a due istanze.

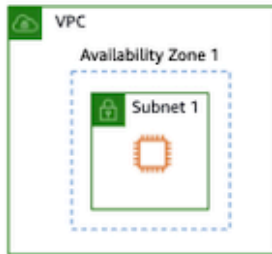
- c. In Sicurezza e rete, scegli Consenti alle applicazioni distribuite in questo ambiente di essere accessibili pubblicamente e scegli almeno due sottoreti pubbliche.
- d. Seleziona Avanti.

# Specify configurations [Info](#)

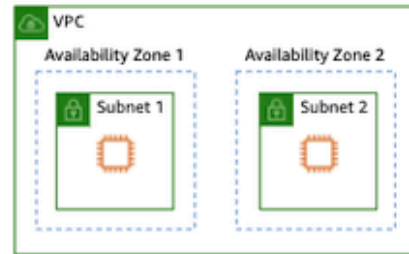
## Availability [Info](#)

Choose the availability pattern for your environment.

- Standalone runtime environment**  
Sets up a single instance in a single availability zone. Does not guarantee high availability but costs less.



- High availability cluster**  
Sets up redundant instances across two availability zones. Enables higher availability but costs more.



## Resources

### Instance type

Choose the instance type for your high availability cluster.

M2.m5.large

### Desired capacity

Specify the desired number of instances.

2

## Security and network

- Allow applications deployed to this environment to be publicly accessible.

### Virtual Private Cloud (VPC)

Choose the VPC where you want to create the environment.

Default vpc-15

### Subnets

Choose one or more subnets for a high availability setup.

Choose subnets

subnet-56f1e | us-west-2a X

subnet-6885 | us-west-2b X

### Security groups

Choose one or more security groups for the chosen VPC.

5. Nella pagina Allega archiviazione, scegli Avanti.
6. Nella pagina Pianifica manutenzione, scegli Nessuna preferenza, quindi scegli Avanti.

**Schedule maintenance** [Info](#)

**Maintenance window** [Info](#)  
Select the period you want pending modifications or maintenance to be applied.

**When to apply modifications**

**No preference**  
AWS will pick an optimized maintenance window for your environment.

**Select new maintenance window**  
Manually set the period you want pending modifications or maintenance to be applied to the operating system and engine version upgrade.

Cancel

7. Nella pagina Rivedi e crea, esamina tutte le configurazioni fornite per l'ambiente di runtime, quindi scegli Crea ambiente.

### Step 3: Attach storage Edit

#### EFS storage

Storage ID	Storage name	Mount point
No storage No storage to display.		

#### FSx storage

Storage ID	Storage name	Mount point
No storage No storage to display.		

### Step 4: Schedule maintenance Edit

#### Maintenance window

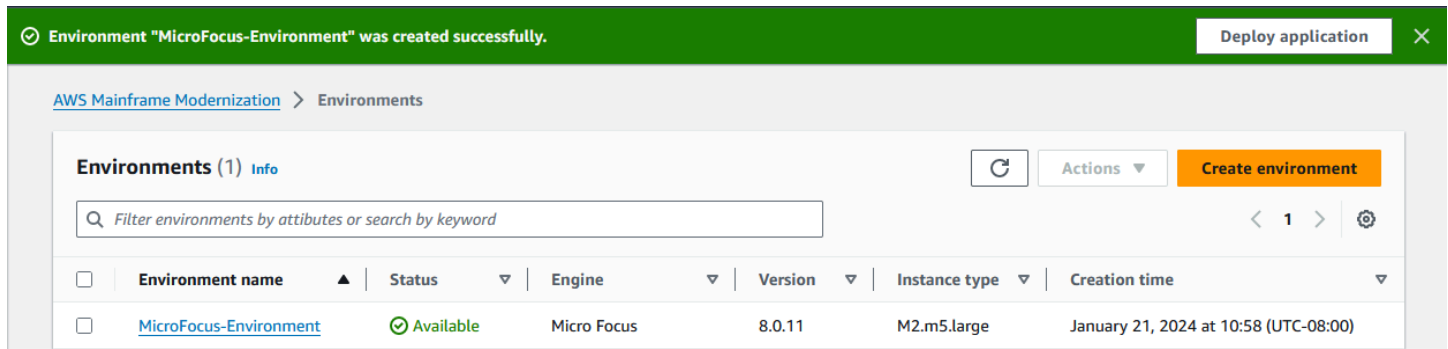
Preferred maintenance window  
No preference

Cancel Previous Create environment

Dopo aver creato l'ambiente, viene visualizzato un banner che dice `Environment name was created successfully`, e il campo Stato diventa Disponibile. Il processo di creazione dell'ambiente richiede alcuni minuti, ma puoi continuare con i passaggi successivi mentre è in esecuzione.

Fase 5: Creare un ambiente di runtime

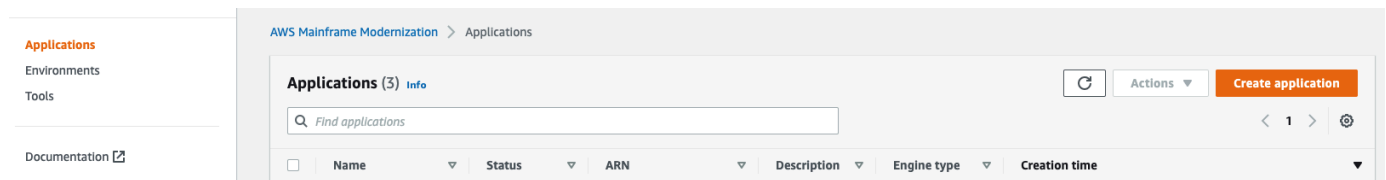
35



## Fase 6: Creare un'applicazione

Per creare un'applicazione

1. Nel riquadro di navigazione, scegliere Applications (Applicazioni). Quindi scegli Crea applicazione.



2. Nella pagina Crea applicazione, in Specificare le informazioni di base, immettete MicroFocus-CardDemo il nome dell'applicazione e in Tipo di motore assicuratevi che Micro Focus sia selezionato. Quindi scegli Successivo.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name

Use only alphanumeric characters, hyphens, and underscores. The maximum length is 60 characters.


Application description - *optional*

*Describe the application*


The maximum length is 500 characters.

### Engine type

**Blu Age**  
This engine provides the framework and dependencies necessary to execute applications refactored by Blu Age.



**Micro Focus**  
This engine provides a mainframe-compatible runtime for replatformed applications by Micro Focus



3. In Specificare risorse e configurazioni, scegliete l'opzione per specificare la definizione dell'applicazione con le relative risorse e configurazioni utilizzando l'editor in linea.

AWS Mainframe Modernization > Applications > Create application

Step 1  
[Specify basic information](#)

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

## Specify resources and configurations [Info](#)

### Resources and configurations

Choose an approach to define the application

- Specify the application definition with its resources and configurations using the inline editor
- Use an application definition JSON file in an Amazon S3 bucket

```
1 {}
```

JSON Ln 1, Col 1 Errors: 0 Warnings: 0

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

Immettete la seguente definizione di applicazione nell'editor:

```
{
  "template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "yourname-aws-region-carddemo",
        "s3-key-prefix": "CardDemo"
      }
    }
  ]
}
```




```

],
"definition": {
  "listeners": [
    {
      "port": 6000,
      "type": "tn3270"
    }
  ],
  "dataset-location": {
    "db-locations": [
      {
        "name": "Database1",
        "secret-manager-arn":
"arn:aws:secretsmanager:Region:123456789012:secret:MicroFocus-Tutorial-RDS-Secret-
xxxxxx"
      }
    ]
  },
  "batch-settings": {
    "initiators": [
      {
        "classes": [
          "A",
          "B"
        ],
        "description": "initiator_AB...."
      },
      {
        "classes": [
          "C",
          "D"
        ],
        "description": "initiator_CD...."
      }
    ],
    "jcl-file-location": "${s3-source}/catalog/jcl"
  },
  "cics-settings": {
    "binary-file-location": "${s3-source}/loadlib",
    "csd-file-location": "${s3-source}/rdef",
    "system-initialization-table": "CARDSIT"
  },
  "xa-resources": [
    {

```

```
    "name": "XASQL",
    "secret-manager-arn":
      "arn:aws:secretsmanager:Region:123456789012:secret:MicroFocus-Tutorial-RDS-Secret-
xxxxxx",
      "module": "${s3-source}/xa/ESPGSQLXA64.so"
  }
]
}
```

 Note

Questo file è soggetto a modifiche.

4. Modifica il JSON dell'applicazione nell'oggetto `properties` di `source-locations` come segue:
  - a. Sostituisci il valore per `s3_bucket` con il nome del bucket Amazon S3 che hai creato nella fase 1.
  - b. Sostituisci il valore per `s3-key-prefix` con la cartella (key prefix) in cui hai caricato i file di `CardDemo` esempio. Se hai caricato la `CardDemo` directory direttamente in un bucket Amazon S3, `s3-key-prefix` non è necessario modificarla.
  - c. Sostituisci entrambi `secret-manager-arn` i valori con l'ARN per il segreto del database creato nel passaggio 4.

## Resources and configurations

Choose an approach to define the application

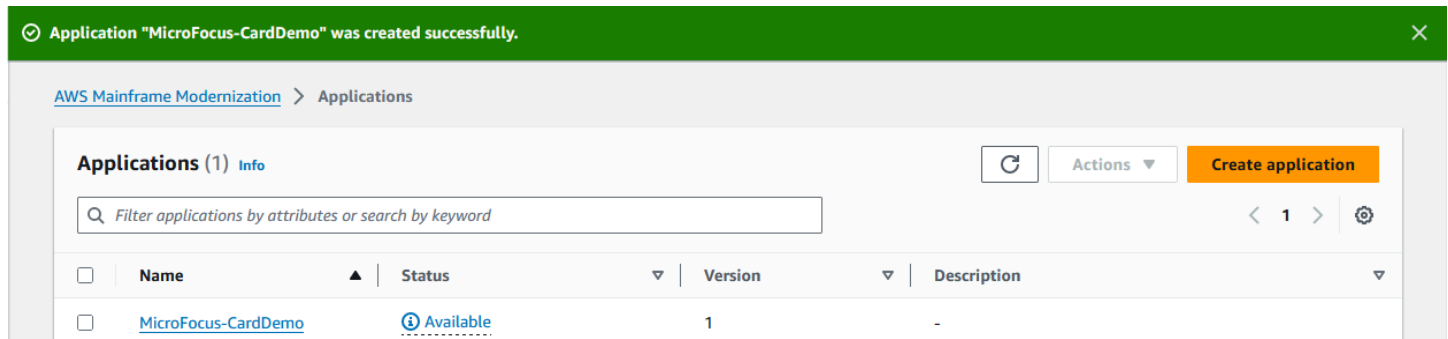
- Specify the application definition with its resources and configurations using the inline editor
- Use an application definition JSON file in an Amazon S3 bucket

```
1 {
2   "template-version": "2.0",
3   "source-locations": [
4     {
5       "source-id": "s3-source",
6       "source-type": "s3",
7       "properties": {
8         "s3-bucket": "XXXXXXXXXXXX-cardemo",
9         "s3-key-prefix": "CardDemo"
10      }
11    }
12  ],
13  "definition": {
14    "listeners": [{"arn": "arn:aws:lambda:us-east-1:123456789012:lambda:function:CardDemo"}],
15    "dataset-location": {
16      "db-locations": [
17        {
18          "name": "Database1",
19          "secret-manager-arn": "arn:aws:secretsmanager:us-east-1:123456789012:secret:XXXXXXXXXXXX"
20        }
21      ]
22    }
23  },
24  "batch-settings": {
25  }
26 }
27 }
28 }
29 }
```

JSON Ln 60, Col 2 ✖ Errors: 0 ⚠ Warnings: 0

Per ulteriori informazioni sulla definizione dell'applicazione, vedere [Definizione dell'applicazione](#) **Micro Focus**.

5. Seleziona Successivo per continuare.
6. Nella pagina Rivedi e crea, esamina le informazioni che hai fornito, quindi scegli Crea applicazione.

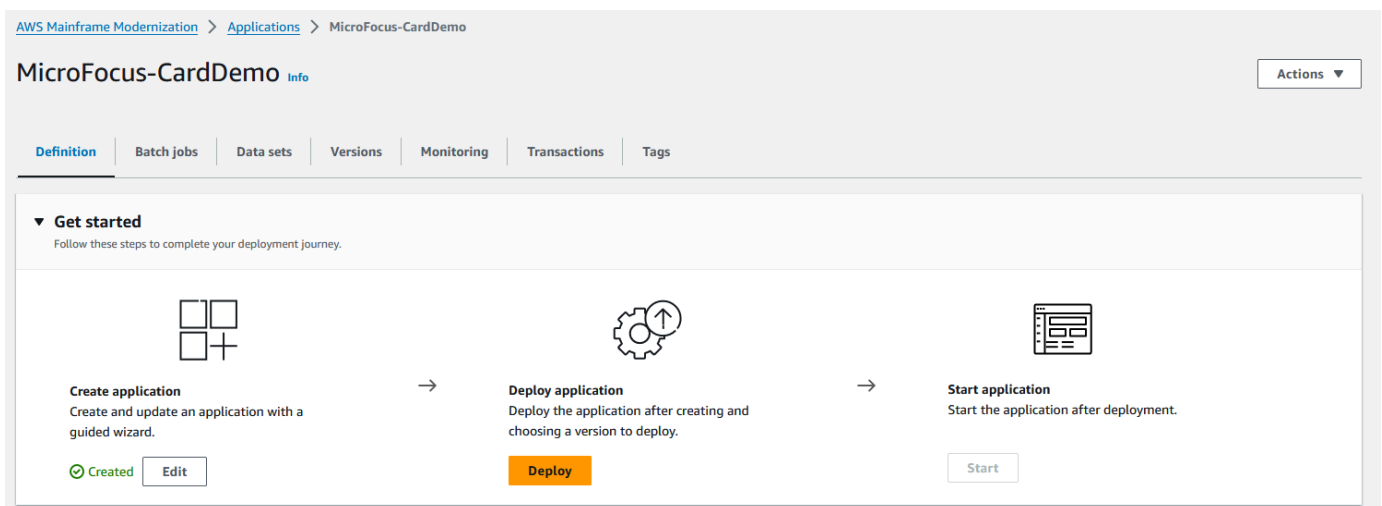


Dopo aver creato l'applicazione, viene visualizzato un banner che dice *Application name* was created successfully. E il campo Stato diventa Disponibile.

## Fase 7: Implementazione di un'applicazione

Per distribuire un'applicazione

1. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli *MicroFocus-CardDemo*.
2. In *Distribuisce applicazione*, scegli *Distribuisce*.



3. Scegli la versione più recente dell'applicazione e dell'ambiente che hai creato in precedenza, quindi scegli *Deploy*.

[AWS Mainframe Modernization](#) > [Applications](#) > [MicroFocus-CardDemo](#) > Deploy application

## Deploy application Info

You have selected the following application:

Name	Description	Engine
MicroFocus-CardDemo	-	Micro Focus

**Available versions (1/1)** ↻

Choose a version from the list.

< 1 > ⚙️

Version
<input checked="" type="radio"/> 1

**Environments (1/1) Info**

< 1 > ⚙️

Environment name	Status	Engine
<input checked="" type="radio"/> <a href="#">MicroFocus-Environment</a>	<span>✔️ Available</span>	Micro Focus

Cancel Deploy

Quando l' CardDemo applicazione viene distribuita correttamente, lo stato diventa Pronto.

✔️ Application "MicroFocus-CardDemo" version 1 has deployed successfully to environment "MicroFocus-Environment". ✕

[AWS Mainframe Modernization](#) > [Applications](#)

**Applications (1) Info** ↻ Actions ▾ Create application

< 1 > ⚙️

<input type="checkbox"/>	Name	Status	Version	Description
<input type="checkbox"/>	<a href="#">MicroFocus-CardDemo</a>	<span>✔️ Ready</span>	1	-

## Fase 8: Importazione dei set di dati

Per importare set di dati

1. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli l'applicazione.
2. Scegli la scheda Set di dati. Scegliere quindi Import (Importa).
3. Scegli Importa e modifica la configurazione JSON, quindi scegli l'opzione Copia e incolla il tuo JSON.

### Import data set [Info](#)

**Choose import method** [Info](#)  
Choose import method.

Import with guided configuration  
Create your own data sets configuration with guidance.

Import and edit JSON configuration  
Use data set configuration JSON files from an Amazon S3 bucket or write your own JSON script.

### JSON configuration

Import from Amazon S3 bucket.

Copy and paste your own JSON.

1		
---	--	--

4. Copia e incolla il seguente codice JSON ma non scegli ancora «Invia». Questo JSON contiene tutti i set di dati richiesti per l'applicazione demo, ma richiede i dettagli del bucket Amazon S3.

```
{
  "dataSets": [
    {
      "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS",
        "relativePath": "DATA",
```

```

        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",
                "primaryKey": {
                    "length": 11,
                    "offset": 0
                }
            }
        },
        "recordLength": {
            "min": 300,
            "max": 300
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS.DAT"
    }
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.CARDDATA.VSAM.AIX.PATH",
        "relativePath": "DATA",
        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",
                "primaryKey": {
                    "length": 11,
                    "offset": 16
                }
            }
        },
        "recordLength": {
            "min": 150,
            "max": 150
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS.DAT"
    }
}

```

```

    },
    {
      "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {
          "vsam": {
            "format": "KS",
            "encoding": "A",
            "primaryKey": {
              "length": 16,
              "offset": 0
            }
          }
        },
        "recordLength": {
          "min": 150,
          "max": 150
        }
      },
      "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDDATA.VSAM.KSDS.DAT"
      }
    },
    {
      "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {
          "vsam": {
            "format": "KS",
            "encoding": "A",
            "primaryKey": {
              "length": 16,
              "offset": 0
            }
          }
        },
        "recordLength": {
          "min": 50,
          "max": 50
        }
      }
    }
  ]
}

```



```

    }
  },
  "externalLocation": {
    "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS.DAT"
  }
},
{
  "dataSet": {
    "storageType": "Database",
    "datasetName": "AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS",
    "relativePath": "DATA",
    "datasetOrg": {
      "vsam": {
        "format": "KS",
        "encoding": "A",
        "primaryKey": {
          "length": 9,
          "offset": 0
        }
      }
    },
    "recordLength": {
      "min": 500,
      "max": 500
    }
  },
  "externalLocation": {
    "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS.DAT"
  }
},
{
  "dataSet": {
    "storageType": "Database",
    "datasetName": "AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH",
    "relativePath": "DATA",
    "datasetOrg": {
      "vsam": {
        "format": "KS",
        "encoding": "A",
        "primaryKey": {
          "length": 11,
          "offset": 25
        }
      }
    }
  }
}

```

```

        }
    },
    "recordLength": {
        "min": 50,
        "max": 50
    }
},
"externalLocation": {
    "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS.DAT"
}
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",
                "primaryKey": {
                    "length": 16,
                    "offset": 0
                }
            }
        },
        "recordLength": {
            "min": 350,
            "max": 350
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT"
    }
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {

```

```
        "vsam": {
            "format": "KS",
            "encoding": "A",
            "primaryKey": {
                "length": 8,
                "offset": 0
            }
        },
        "recordLength": {
            "min": 80,
            "max": 80
        },
        "externalLocation": {
            "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS.DAT"
        }
    }
]
```

5. Sostituisci ogni occorrenza di `<s3-bucket-name>` (ce ne sono otto) con il nome del bucket Amazon S3 che contiene la CardDemo cartella, ad esempio. `your-name-aws-region-carddemo`

#### Note

Per copiare l'URI Amazon S3 per la cartella in Amazon S3, seleziona la cartella, quindi scegli Copia URI Amazon S3.

6. Seleziona Invia.

Al termine dell'importazione, viene visualizzato un banner con il seguente messaggio: Viene visualizzato `Import task with resource identifier name was completed successfully`. un elenco dei set di dati importati.

Import task with resource identifier "1pa6795ukmfr9" was completed successfully.

AWS Mainframe Modernization > Applications > MicroFocus-CardDemo

## MicroFocus-CardDemo Info

Definition | Batch jobs | **Data sets** | Versions | Monitoring | Transactions | Tags

**Data sets (8) Info** Last updated (UTC-08:00)  
January 24, 2024, 15:25 ↻ Import history Import

Filter data sets by name

Data set name	Data set org	Format
<a href="#">AWS.M2.CARDDemo.ACCTDATA.VSAM.KSDS</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.CARDDATA.VSAM.AIX.PATH</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.CARDDATA.VSAM.KSDS</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.CARDXREF.VSAM.AIX.PATH</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.CARDXREF.VSAM.KSDS</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.CUSTDATA.VSAM.KSDS</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.TRANSACT.VSAM.KSDS</a>	VSAM	KS
<a href="#">AWS.M2.CARDDemo.USRSEC.VSAM.KSDS</a>	VSAM	KS

Puoi anche visualizzare lo stato di tutte le importazioni di set di dati scegliendo Cronologia importazioni nella scheda Set di dati.

## Fase 9: Avviare un'applicazione

Per avviare un'applicazione


1. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli l'applicazione.
2. Scegli Avvia applicazione.

AWS Mainframe Modernization > Applications > MicroFocus-CardDemo

## MicroFocus-CardDemo Info


Definition | Batch jobs | Data sets | Versions | Monitoring | Transactions | Tags

**Get started**  
Follow these steps to complete your deployment journey.




**Create application**  
Create and update an application with a guided wizard.

✔ Created Edit



**Deploy application**  
Version 1 of MicroFocus-CardDemo has been deployed.

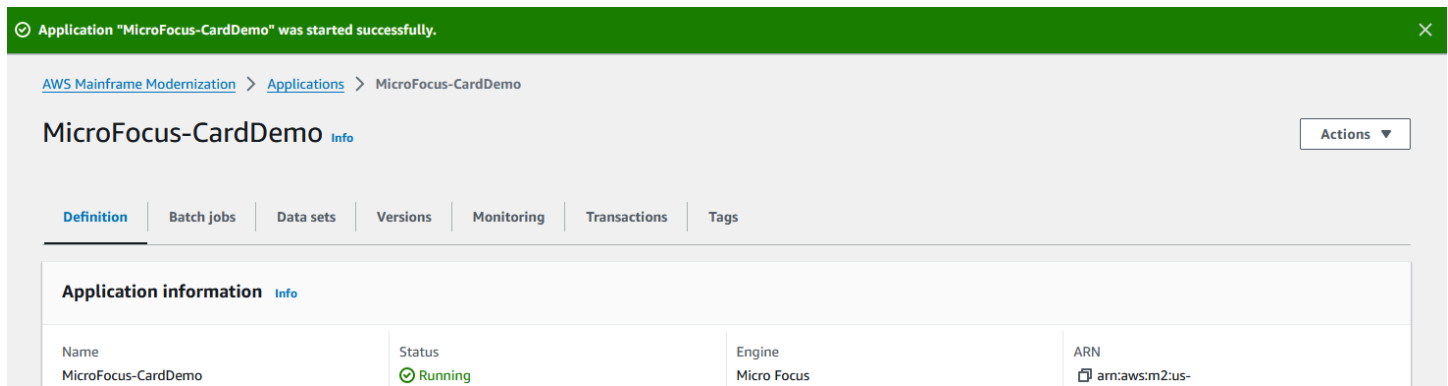
✔ Deployed Deploy



**Start application**  
Start the application after deployment.

⊖ Stopped Start

Quando l' CardDemo applicazione inizia a funzionare correttamente, viene visualizzato un banner con il seguente messaggio: `Application name was started successfully`. Il campo Stato diventa In esecuzione.



## Fase 10: Connect all'applicazione CardDemo CICS

Prima di connetterti, assicurati che il VPC e il gruppo di sicurezza che hai specificato per l'applicazione siano gli stessi che hai applicato per l'interfaccia di rete da cui ti conatterai.

Per configurare la connessione TN3270, sono necessari anche il nome host DNS e la porta dell'applicazione.

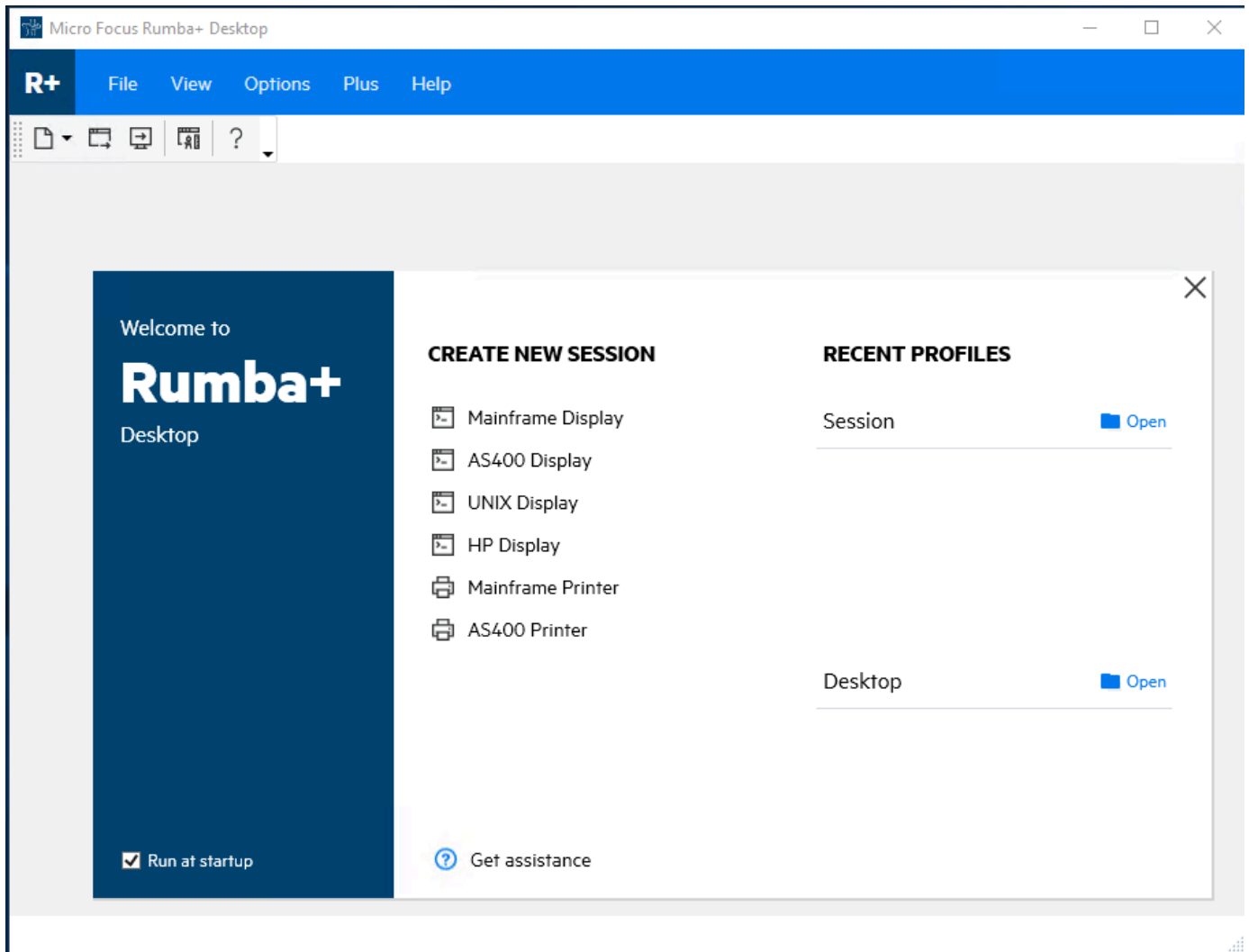
Per configurare e connettere un'applicazione al mainframe utilizzando l'emulatore di terminale

1. Apri la console di modernizzazione del AWS mainframe e scegli Applicazioni, quindi scegli. MicroFocus-CardDemo
2. Scegli l'icona di copia per copiare il nome host DNS. Assicurati anche di annotare il numero di porte.
3. Avvia un emulatore di terminale. Questo tutorial utilizza Micro Focus Rumba+.

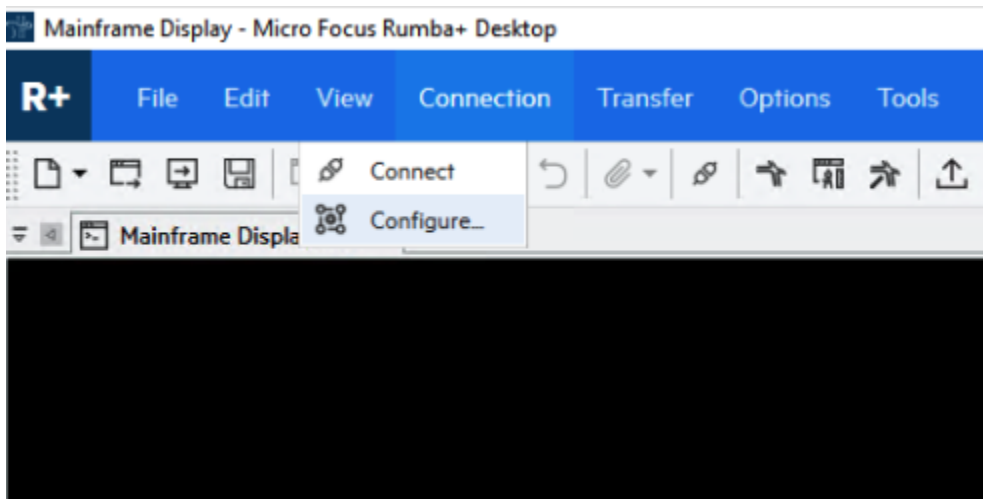
### Note

I passaggi di configurazione variano a seconda dell'emulatore.

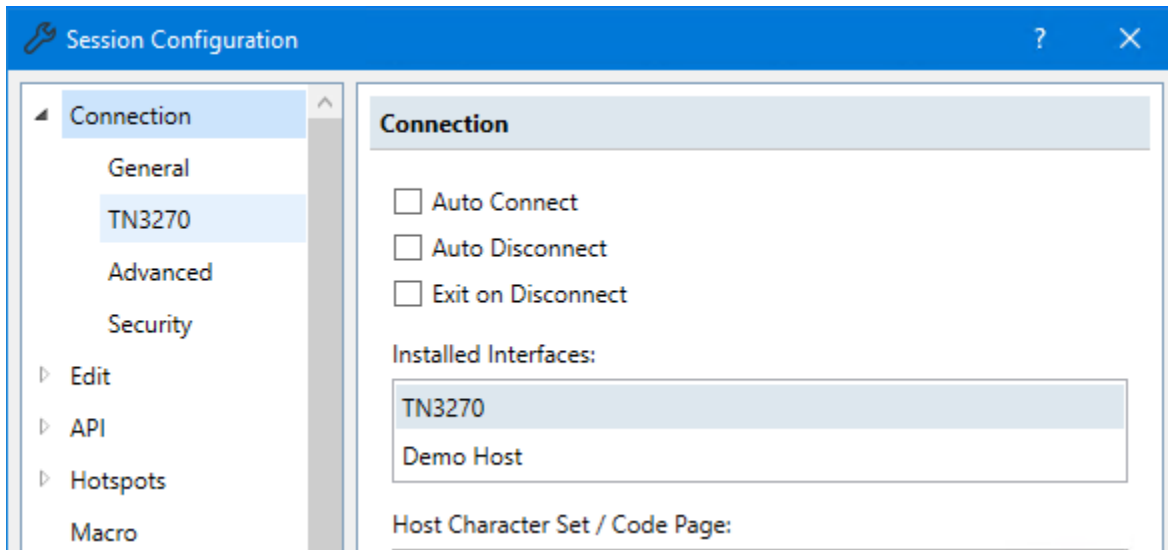
4. Scegli Mainframe Display.



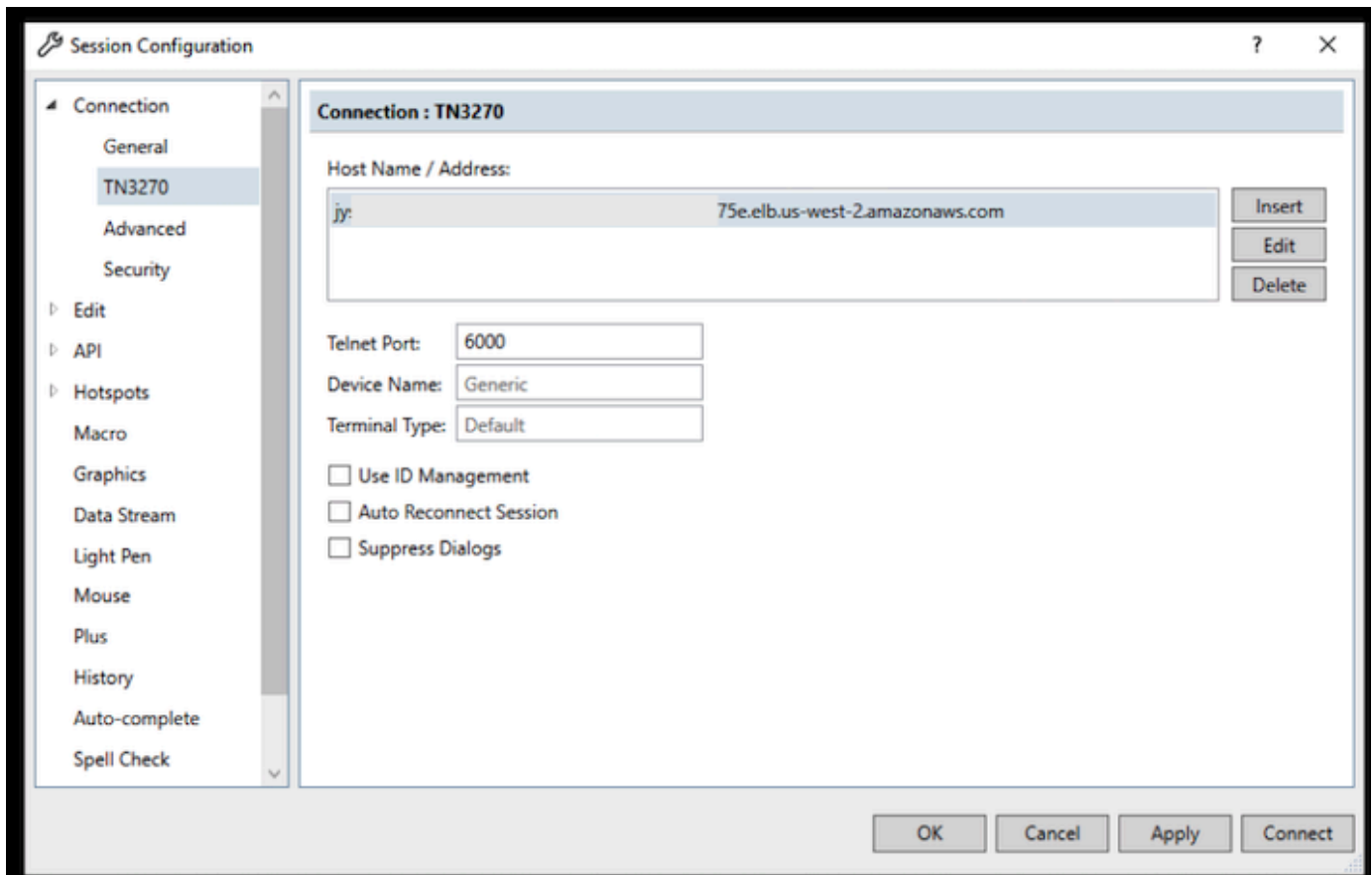
5. Scegli Connessione, quindi scegli Configura.




6. In Interfacce installate, scegliete **TN3270**, quindi scegliete **TN3270** nuovamente nel menu Connessione.




7. Scegliete Inserisci e incollate il file DNS Hostname per l'applicazione. 6000 Specificate la porta Telnet.



 Note

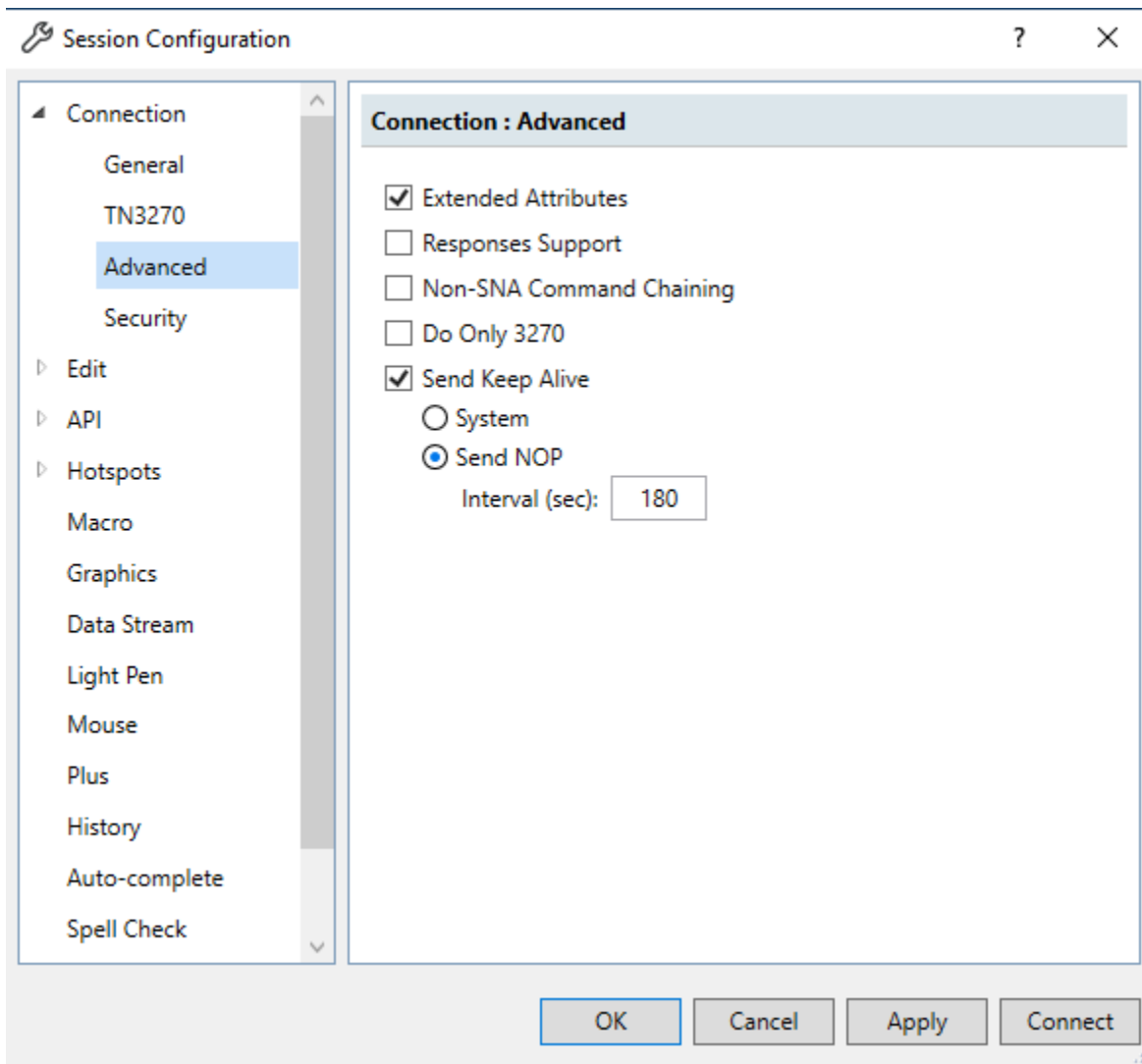
Se utilizzi AWS AppStream 2.0 in un browser e hai difficoltà a incollare i valori, consulta [Risoluzione dei problemi degli utenti della AppStream versione 2.0](#).

8. In Connessione, scegli Advanced, quindi scegli Send Keep Alive e Send NOP e inserisci 180 per l'intervallo.

 Note

La configurazione dell'impostazione keep alive sul terminale TN3270 su almeno 180 secondi aiuta a garantire che il Network Load Balancer non interrompa la connessione.





## 9. Scegli Connetti.

### Note

Se la connessione fallisce:

- Se utilizzi la AppStream versione 2.0, verifica che il VPC e il gruppo di sicurezza specificati per l'ambiente dell'applicazione siano gli stessi della flotta 2.0. AppStream
- Usa il VPC Reachability Analyzer per analizzare la connessione. [È possibile accedere al Reachability Analyzer tramite la console.](#)
- Come fase di diagnostica, prova ad aggiungere o modificare le regole in entrata del gruppo di sicurezza per l'applicazione in modo da consentire il traffico verso la

porta 6000 da qualsiasi luogo (ad esempio CIDR Block 0.0.0.0/0). Se ti connetti con successo, allora sai che il gruppo di sicurezza stava bloccando il tuo traffico. Cambia l'origine del gruppo di sicurezza con qualcosa di più specifico. Per ulteriori informazioni sui gruppi di sicurezza, consulta Nozioni di [base sui gruppi di sicurezza](#).

10. Inserisci USER0001 il nome utente e password la password.

### Note

In Rumba, l'impostazione predefinita per Clear è e ctrl-shift-z l'impostazione predefinita per Reset è ctrl-r.

```

Mainframe Display - Micro Focus Rumba+ Desktop
R+ File Edit View Connection Transfer Options Tools Plus Help
Mainframe Display
Tran : CC00          AWS Mainframe Modernization      Date : 01/22/24
Prog : CDSGN00C     CardDemo                                           Time : 00:00:49
AppID: SBP7CMEZ                                         SysID: CARD

This is a Credit Card Demo Application for Mainframe Modernization

+=====+
|%%%%%%%% NATIONAL RESERVE NOTE %%%%%%%%%|
|%(1) THE UNITED STATES OF KICSLAND (1)%|
|%%$%          ***** %%$%|
|%%$ {x}          (o o)          %%$|
|%%$ ***** ( V )          ONE %%$|
|%(1)          ---m-m---          (1)%|
|%%~~~~~ ONE DOLLAR ~~~~~%|
+=====+

Type your User ID and Password, then press ENTER:

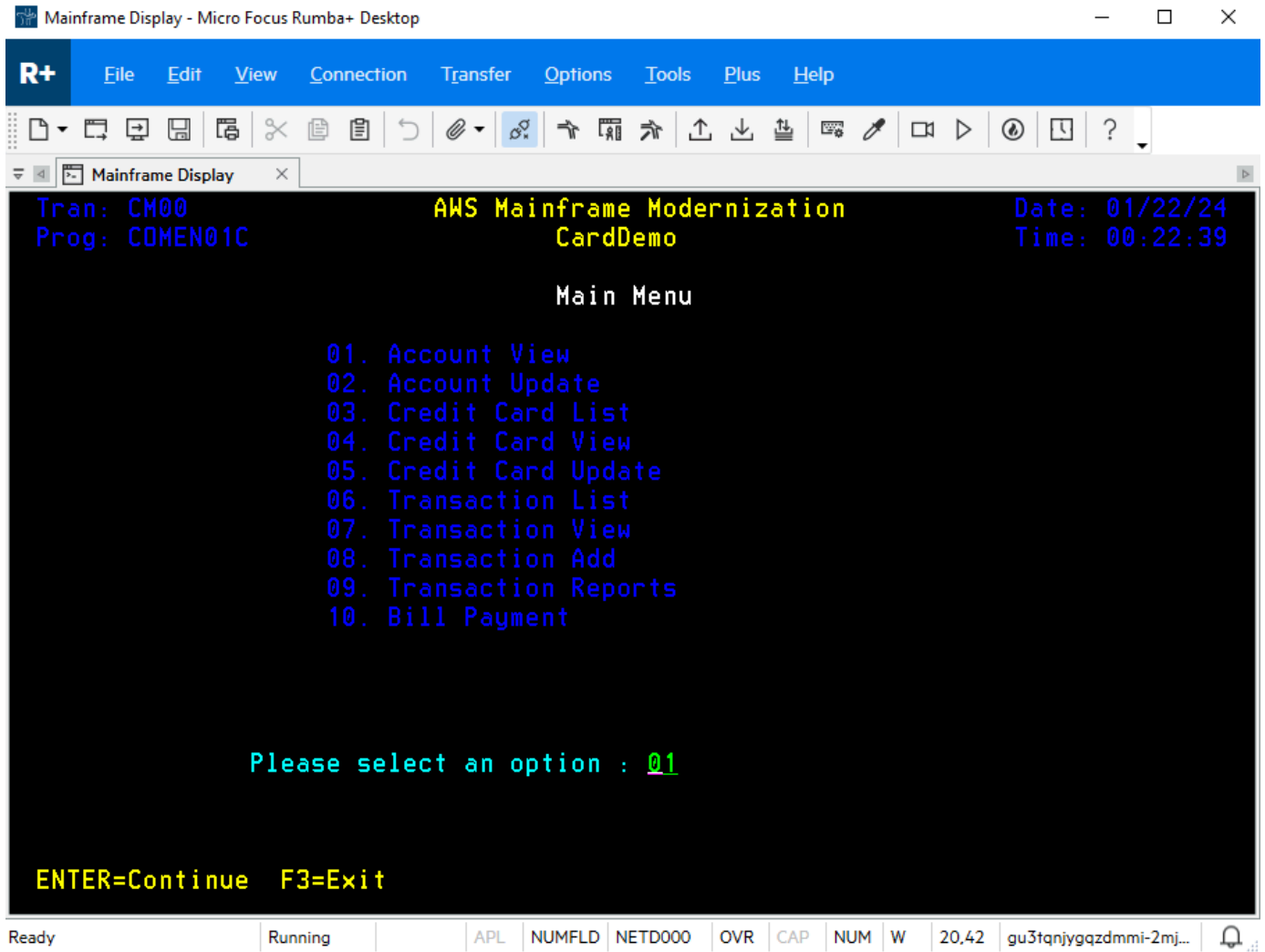
User ID      : user0001 (8 Char)
Password     :          (8 Char) _

ENTER=Sign-on  F3=Exit

Ready | Running | APL | NUMFLD | NETB000 | OVR | CAP | NUM | W | 20.62 | gu3tqnjyqzdmimi-2mj...

```

11. Dopo aver effettuato correttamente l'accesso, puoi navigare all'interno dell' CardDemoapplicazione.
12. Accedi 01 alla visualizzazione dell'account.



The screenshot shows a terminal window titled "Mainframe Display - Micro Focus Rumba+ Desktop". The window has a blue menu bar with "R+" and options like File, Edit, View, Connection, Transfer, Options, Tools, Plus, and Help. Below the menu bar is a toolbar with various icons. The main display area has a black background with yellow and blue text. At the top, it shows "Tran: CM00", "Prog: COMEN01C", "AWS Mainframe Modernization", "CardDemo", "Date: 01/22/24", and "Time: 00:22:39". Below this is a "Main Menu" with a list of options: 01. Account View, 02. Account Update, 03. Credit Card List, 04. Credit Card View, 05. Credit Card Update, 06. Transaction List, 07. Transaction View, 08. Transaction Add, 09. Transaction Reports, and 10. Bill Payment. At the bottom of the menu, it says "Please select an option : 01" and "ENTER=Continue F3=Exit". At the very bottom of the window, there is a status bar with "Ready", "Running", and various system parameters like APL, NUMFLD, NETD000, OVR, CAP, NUM, W, 20.42, and gu3tanjyqazdmmi-2mj...

13. Inserisci 0000000010 il numero di conto e premi Invio sulla tastiera.

#### Note

Altri account validi sono 0000000011 e 0000000020.

The screenshot shows a terminal window titled 'Mainframe Display - Micro Focus Rumba+ Desktop'. The window contains the following text:

```

Tran: CAVW          AWS Mainframe Modernization      Date: 01/22/24
Prog: COACTVWC     CardDemo                                           Time: 00:24:48

                          View Account
                          Account Number : 00000000010      Active Y/N: Y
Opened: 2015-09-13   Credit Limit          : + 5,401.00
Expiry: 2023-01-27   Cash credit Limit     : + 4,442.00
Reissue: 2023-01-27   Current Balance       : + 159.00
                          Current Cycle Credit: + .00
Account Group: _____ Current Cycle Debit : + .00

                          Customer Details
Customer id : 000000010      SSN: 754-75-5746
Date of birth: 1980-06-11   FICO Score: 476
First Name   Middle Name:   Last Name :
Maubell     Creola         Mann
Address: 77933 Adah Dale      State      CT
          Suite 343          Zip        44803
City        Andersonfurt    Country    USA
Phone 1: (614)594-2619   Government Issued Id Ref : 00000000000212824755
Phone 2: (667)057-0235   EFT Account Id: 0093803568 Primary Card Holder Y/N: Y

                          Enter or update id of account to display

F3=Exit
  
```

At the bottom of the window, there is a status bar with the following information: Ready | Running | APL | NUMFLD | NETD000 | OVR | CAP | NUM | W | 5.39 | gu3tqnjyqzdmimi-2mj... | [notification icon]

14. Premi F3 per uscire dal menu e F3 per uscire dalla transazione.

## Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale per evitare costi aggiuntivi. Per fare ciò, completa la seguente procedura:

- Se necessario, interrompete l'applicazione.
- Elimina l'applicazione . Per ulteriori informazioni, consulta [Eliminare un'applicazione di modernizzazione AWS del mainframe](#).
- Eliminare l'ambiente di runtime. Per ulteriori informazioni, consulta [Eliminare un ambiente di runtime di AWS modernizzazione del mainframe](#).

- Elimina i bucket Amazon S3 che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminazione di un bucket nella Guida](#) per l'utente di Amazon S3.
- Elimina il AWS Secrets Manager segreto che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminare un segreto](#).
- Elimina la chiave KMS che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminazione delle chiavi AWS KMS](#).
- Elimina il database Amazon RDS che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminare l'istanza EC2 e l'istanza DB](#) nella Amazon RDS User Guide.
- Se hai aggiunto una regola del gruppo di sicurezza per la porta 6000, elimina la regola.

## Passaggi successivi

Per informazioni su come configurare un ambiente di sviluppo per le applicazioni modernizzate, consultate [Tutorial: Configurazione AppStream 2.0 per l'uso con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#).

# Approccio alla modernizzazione

La migrazione è complessa e presenta molte variabili. AWS La modernizzazione del mainframe offre un approccio evolutivo che offre alcuni vantaggi a breve termine migliorando l'agilità con numerose opportunità di ottimizzazione e innovazione in seguito. Inoltre, la modernizzazione del AWS mainframe aiuta a semplificare il percorso e rispetta comunque i dettagli dell'azienda e del business dei clienti. I due approcci principali supportati da AWS Mainframe Modernization sono il refactoring automatico o il replatforming. Quale scegliere dipende dalla situazione del cliente.

Il refactoring automatizzato utilizza gli strumenti AWS Blu Age per convertire automaticamente codice, dati e dipendenze in linguaggio, datastore e framework moderni, garantendo allo stesso tempo l'equivalenza funzionale con le stesse funzioni aziendali.

Replatforming utilizza gli strumenti Micro Focus per trasformare i carichi di lavoro mainframe in servizi agili. AWS

Puoi pensare al percorso di modernizzazione per fasi. La prima fase comprende tre fasi: valutazione, mobilitazione, migrazione e modernizzazione. La fase successiva include la fase di gestione e ottimizzazione, in cui è possibile identificare maggiori opportunità di innovazione.

## Argomenti

- [Fase di valutazione](#)
- [Fase di mobilitazione](#)
- [Fase di migrazione e modernizzazione](#)
- [Fase operativa e di ottimizzazione](#)

## Fase di valutazione

Al livello più alto, la fase di valutazione verifica se sei pronto per la migrazione. Definite un business case e poi istruite il vostro team con workshop e una giornata di immersione (demo e laboratori) offerti da AWS I workshop e le giornate di immersione affrontano argomenti diversi. Queste attività vengono svolte al di fuori della modernizzazione del AWS mainframe.

## Fase di mobilitazione

Nella fase Mobilize, si inizia il progetto con un avvio, quindi si esegue un processo di scoperta che estrae i dati dalle applicazioni mainframe e li inserisce in uno strumento di migrazione. Identificate

le applicazioni da migrare e selezionate alcune applicazioni da pilotare. Sarete voi a perfezionare il vostro business case, redigere il piano di migrazione e decidere come gestire la sicurezza e la conformità, la governance degli account e il modello operativo. Hai creato un centro cloud di eccellenza con le persone giuste del tuo team. Dirigi i progetti pilota e documenti ciò che hai imparato. Perfezionate il vostro piano di migrazione e il vostro business case. Molte di queste attività vengono svolte al di fuori della modernizzazione del AWS mainframe.

## Fase di migrazione e modernizzazione

La fase di migrazione e modernizzazione si applica a ciascuna applicazione e consiste in diverse attività, tra cui l'assegnazione di persone, l'esecuzione di attività di discovery approfondite, l'individuazione della giusta architettura applicativa, la configurazione degli ambienti di runtime delle applicazioni AWS, la riplatforma o il refactoring del codice, l'integrazione con altri sistemi e, ovviamente, i test. Al termine della fase, distribuisce le applicazioni riplatformate o rifattorizzate in produzione e passerai al nuovo sistema su AWS. La maggior parte o tutte queste attività vengono svolte in AWS Mainframe Modernization, in un altro AWS servizio o in uno strumento a cui Mainframe Modernization fornisce l'accesso. AWS

[Se desideri utilizzare il refactoring automatico, consulta Blu Insights.](#) AWS Blu Insights è ora disponibile AWS Management Console tramite Single Sign-On. Non è più necessario gestire credenziali AWS Blu Insights separate. È possibile accedere alle funzionalità di AWS AWS Blu Age Codebase e Transformation Center direttamente da AWS Management Console

Per la migrazione dei dati dal mainframe a AWS, consigliamo il e il AWS SCT. AWS Database Migration Service Per ulteriori informazioni, consulta [What is the AWS Schema Conversion Tool?](#) nella AWS Schema Conversion Tool User Guide e [Cos'è AWS Database Migration Service?](#) nella Guida AWS Database Migration Service per l'utente.

## Fase operativa e di ottimizzazione

Nella fase Operate and Optimize, ci si concentra sul monitoraggio delle applicazioni distribuite, sulla gestione delle risorse e sulla garanzia che la sicurezza e la conformità siano aggiornate. Valuterai inoltre le opportunità per ottimizzare i carichi di lavoro migrati.

# Concetti

AWS La modernizzazione del mainframe fornisce strumenti e risorse per aiutarti a migrare, modernizzare ed eseguire carichi di lavoro mainframe su. AWS

## Argomenti

- [Applicazione](#)
- [Definizione dell'applicazione](#)
- [Lavoro in batch](#)
- [Configurazione](#)
- [Set di dati](#)
- [Ambiente](#)
- [Modernizzazione del mainframe](#)
- [Viaggio di migrazione](#)
- [Punto di montaggio](#)
- [Rifattorizzazione automatizzata](#)
- [Ridefinizione della piattaforma](#)
- [Risorsa](#)
- [Motore di runtime](#)

## Applicazione

Un carico di lavoro mainframe in esecuzione in Mainframe Modernization. AWS Un'applicazione è costituita da un insieme di processi in batch, transazioni interattive (CICS o IMS) o altri componenti. L'ambito è definito dall'utente. È necessario definire e specificare tutti i componenti o le risorse necessari per il carico di lavoro, ad esempio transazioni CICS o processi batch.

## Definizione dell'applicazione

La definizione o la specifica dei componenti e delle risorse necessari a un'applicazione (carico di lavoro mainframe) in esecuzione in AWS Mainframe Modernization. Separare la definizione dall'applicazione stessa è importante perché è possibile riutilizzare la stessa definizione per più fasi (pre-produzione, produzione), rappresentate da diversi ambienti di runtime.



## Lavoro in batch

Un programma pianificato configurato per essere eseguito senza richiedere l'interazione dell'utente. In AWS Mainframe Modernization, dovrai archiviare sia i file JCL dei processi batch che i file binari dei processi batch in un bucket Amazon S3 e fornire la posizione di entrambi nel file di definizione dell'applicazione. Quando esegui un processo in batch, AWS Mainframe Modernization riporta i seguenti valori di stato:

### Inviando

Il processo batch è in fase di invio.

### Sospensione

Il processo batch è sospeso.

### Spedizione

Il processo batch è in fase di invio.

### In esecuzione

Il processo batch è attualmente in esecuzione.

### Annullamento in corso

Il processo batch è in fase di annullamento.

### Annullato

Il processo batch è stato annullato.

### Riuscito

L'esecuzione del processo batch è stata completata correttamente.

### Non riuscito

Il processo batch non è riuscito.

### Riuscito con avviso

L'esecuzione del processo batch è stata completata correttamente con un errore minore segnalato. Il codice della condizione del lavoro restituito come parte della GetBatchJobExecution risposta indica la causa dell'errore.

## Configurazione

Le caratteristiche di un ambiente o di un'applicazione. Le configurazioni dell'ambiente comprendono il tipo di motore, la versione del motore, i modelli di disponibilità, le configurazioni opzionali del file system e altro ancora.

Le configurazioni delle applicazioni possono essere statiche o dinamiche. Le configurazioni statiche cambiano solo quando si aggiorna un'applicazione distribuendo una nuova versione. Le configurazioni dinamiche, che in genere sono un'attività operativa come l'attivazione o la disattivazione del tracciamento, cambiano non appena vengono aggiornate.

## Set di dati

Un file contenente dati utilizzabili dalle applicazioni.

## Ambiente

Una combinazione denominata di risorse di AWS elaborazione, un motore di runtime e dettagli di configurazione creata per ospitare una o più applicazioni.

## Modernizzazione del mainframe

Il processo di migrazione delle applicazioni da un ambiente mainframe legacy a AWS.

## Viaggio di migrazione

Il end-to-end processo di migrazione e modernizzazione delle applicazioni legacy, in genere composto dalle seguenti fasi: valutazione, mobilitazione, migrazione e modernizzazione, gestione e ottimizzazione.

## Punto di montaggio

Una directory in un file system che fornisce l'accesso ai file archiviati all'interno di tale sistema.

## Rifattorizzazione automatizzata

Il processo di modernizzazione degli artefatti delle applicazioni legacy per l'esecuzione in un ambiente cloud moderno. Può includere la conversione di codice e dati. Per ulteriori informazioni, vedere [AWS Mainframe Modernization Automated Refactor](#).

## Ridefinizione della piattaforma

Processo di spostamento di un'applicazione e degli elementi applicativi da una piattaforma di elaborazione a una piattaforma di elaborazione diversa. Per ulteriori informazioni, consulta [AWS Mainframe Modernization Replatform](#)

## Risorsa

Un componente fisico o virtuale all'interno di un sistema informatico.

## Motore di runtime

Software che facilita l'esecuzione di un'applicazione.

# Refattorizzazione automatica delle applicazioni con AWS Blu Age

Il refactoring automatizzato con AWS Blu Age offre una end-to-end soluzione per la migrazione e la modernizzazione delle applicazioni mainframe. Le fasi del processo di refactoring sono le seguenti:

- Analizza l'inventario
- Analizza le dipendenze
- Trasforma automaticamente il codice
- Acquisisci e gestisci scenari di test

Puoi completare i passaggi precedenti nello strumento Blu Insights, disponibile tramite Single Sign-On dalla console di modernizzazione del AWS mainframe. [Per ulteriori informazioni su Blu Insights, consulta la documentazione di Blu Insights.](#)

Quando sei soddisfatto del codice sorgente trasformato, è il momento di passare a AWS, dove completerai i seguenti passaggi:

- Crea e distribuisce l'applicazione refactorizzata.
- Implementa e monitora la tua applicazione in Mainframe Modernization. AWS

AWS Blu Age Runtime (non gestito) è una delle offerte del servizio di modernizzazione del AWS mainframe insieme a Blu Age managed. AWS Con AWS Blu Age managed, puoi implementare la tua applicazione modernizzata in un ambiente AWS gestito che semplifica la tua esperienza, in modo da non dover gestire l'infrastruttura sottostante che esegue l'applicazione modernizzata. Al contrario, con AWS Blu Age Runtime (non gestito) puoi distribuire la tua applicazione modernizzata nel tuo AWS account, in modo da poter gestire la tua infrastruttura. Con AWS Blu Age Runtime (non gestito) avete la flessibilità necessaria per utilizzare tutti i componenti tecnici necessari per eseguire la vostra applicazione modernizzata nel modo desiderato.

AWS Blu Age Runtime (non gestito) è disponibile per la distribuzione su Amazon EC2 e su Amazon ECS su. AWS Fargate

## Argomenti

- [AWS Note di rilascio di Blu Age](#)

- [AWS I concetti di Blu Age Runtime](#)
- [AWS File di configurazione e configurazione di Blu Age Runtime](#)
- [AWS API Blu Age Runtime](#)
- [AWS Configurazione Blu Age Runtime \(non gestita\)](#)
- [Modifica il codice sorgente con Blu Age Developer IDE](#)

## AWS Note di rilascio di Blu Age

Questa sezione contiene le note di rilascio di AWS Blu Age Runtime e Modernization Tools dalla versione 3.5.0 in poi, prima la più recente, organizzate per numero di versione.

### Note

Per le note di rilascio precedenti a questo documento, contatta i servizi di consegna AWS Blu Age. Per informazioni sulle ultime funzionalità di Blu Insights, consulta le [versioni di Blu Insights](#).

### Argomenti

- [Note di rilascio 3.10.0](#)
- [Versione di runtime 3.10.0](#)
- [Versione 3.10.0 degli strumenti di modernizzazione](#)
- [Note di rilascio 3.9.0](#)
- [Versione di runtime 3.9.0](#)
- [Versione 3.9.0 degli strumenti di modernizzazione](#)
- [Note di rilascio 3.8.0](#)
- [Versione di runtime 3.8.0](#)
- [Versione 3.8.0 degli strumenti di modernizzazione](#)
- [Note di rilascio 3.7.0](#)
- [Versione di runtime 3.7.0](#)
- [Versione 3.7.0 degli strumenti di modernizzazione](#)
- [Note di rilascio 3.6.0](#)

- [Versione di runtime 3.6.0](#)
- [Versione 3.6.0 degli strumenti di modernizzazione](#)
- [Note di rilascio 3.5.0](#)
- [Versione di runtime 3.5.0](#)
- [Versione 3.5.0 degli strumenti di modernizzazione](#)

## Note di rilascio 3.10.0

Questa versione di AWS Blu Age Runtime and Modernization Tools si concentra sugli aggiornamenti e sui miglioramenti di base del prodotto, con l'obiettivo di aumentare le prestazioni e la robustezza in tutte le fasi di trasformazione ed esecuzione. Alcune caratteristiche e modifiche chiave di questa versione sono:

- Aggiornamento della versione da Java 8 a Java 17, che aumenta la sicurezza e le prestazioni e consente ai clienti di distribuire ed eseguire applicazioni implementate in un linguaggio più moderno e di utilizzare versioni recenti di framework di terze parti.
- Supporto aggiuntivo per la gestione di ampi spazi di memoria condivisi tra utenti o processi, l'archiviazione dei dati riutilizzabili dopo il riavvio dell'applicazione o dell'istanza.
- Accesso più rapido a set di dati di grandi dimensioni in Blusam utilizzando un meccanismo di impaginazione che consente di recuperare un sottoinsieme di record in modo incrementale.

Per ulteriori informazioni sulle modifiche incluse in questa versione, consultate le seguenti sezioni.

## Versione di runtime 3.10.0

Questo runtime è basato su Java17, Spring2.7 e Angular16.

### zOS

#### Nuove funzionalità

- Blusam - Aggiunto il supporto per set di dati di grandi dimensioni attraverso un meccanismo impaginato in cui gli indici vengono archiviati e caricati utilizzando le pagine

#### Miglioramenti

- DataUtils.compare migliorato per gestire la conversione con precedenza inferiore da stringa a numero
- È stato aggiunto il supporto per verificare che non ByteRange venga creato alcun valore con valori impropri tramite la proprietà YML DataSimplifier. byteRangeBoundsControlla
- RemoveSosi () migliorato per supportare l'inizializzazione di un GraphicAlphanumericType con un carattere vuoto
- Maggiore robustezza per il funzionamento del lavoro e lettura sicura dello stato GDG
- Blusam - Aggiunto il supporto per la cancellazione di Ehcach dei set di dati Blusam tramite un nuovo metodo denominato .removeCache () CoreBluesamManager
- Blusam - Comportamento migliorato di eliminazione/ridenominazione per i normali set di dati Blusam
- Redis - Supporto migliorato per lo sblocco dei set di dati e la cancellazione del blocco dei record
- JICS - Migliorato il messaggio di errore per le richieste non riuscite
- JCL - Aggiunto il supporto per la concatenazione di variabili ControlM basata sul carattere del punto
- JCL - Aggiunto il supporto per Write ADVANCING (ADV) per i file GDG
- JCL - Supporto migliorato per il numero di generazione corrente dopo l'eliminazione di tutti i file GDG
- JCL - Supporto migliorato per la lettura RDW/RecordSize dal catalogo durante la creazione del set di dati
- JCL - Aggiunto il supporto per l'aggiornamento dell'oggetto risorsa (da AbstractSequentialFile) all'apertura del file con la dimensione del record di output dei dati
- JCL - Prestazioni IDCAMS migliorate
- JCL - Supporto migliorato per PRINT STATEMENT aggiungendo «CHAR» come alias di «CHARACTER»
- SORT - Supporto migliorato per le operazioni di copia da un set di dati a lunghezza fissa Blusam a un set di dati a lunghezza variabile
- SORT - Grammatica di ordinamento migliorata per gestire alcune istruzioni specifiche

## AS400

### Nuove funzionalità

- Aggiunto il supporto per User Spaces e le relative API

- È stato aggiunto il supporto per il parametro TOMSGQ di SNDPGMMSG e le code di messaggi implementate
- CL - Aggiunto il supporto per i parametri FILE e SPLFNAME per il comando OVRPRTF
- CL - Aggiunto il supporto per la gestione delle librerie per la tabella delle partizioni corrispondente con il comando CPYF
- CL - Aggiunto il supporto per la gestione del comando CHGCURLIB e la considerazione della libreria corrente durante la creazione di query
- CL - Aggiunto il supporto per la gestione del comando cl come parte della chiamata stacktrace

## Miglioramenti

- Migliorato MessageHandlingBuilder per una migliore gestione dell'immissione della traccia dello stack di chiamate
- Esecuzione parallela migliorata della funzionalità ContextPreConstruct
- Attributi di visualizzazione migliorati quando un record viene creato da SFLINZ
- SAVOBJ migliorato per consentire la gestione di più file di output
- Migliore gestione dei programmi groovy aggiungendoli programCallStack quando vengono chiamati da un programma Java
- Migliore rilevamento del posizionamento in alto della modalità di aiuto
- Funzionalità TopGMQ migliorata quando viene fornito il parametro TomSGQ per SNDPGMMSG
- Migliore recupero dei messaggi predefiniti e funzionalità del caricatore di messaggi
- Migliore gestione CPYTOIMPF dei caratteri delimitatori nel contenuto
- Blocco del rilascio migliorato sul record READ

## Capacità trasversali

### Nuove funzionalità

- Aggiunta una traduzione per i messaggi di sistema sul Front-End
- Aggiunto un nuovo metodo ExecutionContext per restituire lo stack di chiamate al programma
- Imposta un separatore di riga (per il semplificatore di dati) indipendentemente dall'ambiente effettivo
- Aggiunta la possibilità di configurare il percorso JSON del modello SQL



## Miglioramenti

- Migliorato il metodo di confronto DataUtils. compareAlphaInt() quando si tratta di imbottitura
- Creazione di un flag per consentire un comportamento personalizzato in caso di eccezioni nelle query con il cursore
- Conversione grafica migliorata del db LOWVALUES

## Terza parte

- Aggiornamento per mitigare CVE-2024-21634, CVE-2023-34055, CVE-2023-34462, IN1-JAVA-ORGSRINGFRAMEWORKSECURITY-5905484, CVE-2023-46120, CVE-2023-6481, CVE-2023-6481, CVE-2023-6378, CVE-2023-5072)

## Versione 3.10.0 degli strumenti di modernizzazione

### ZoS

#### Miglioramenti

- COBOL - Aggiunto il supporto per la funzione ABS
- JCL - Ambito variabile avanzato: collegato a STEP anziché a JOB
- Iniezione migliorata dei parametri del cursore per valori bassi/alti
- Analisi CSD migliorata, in particolare per le TRANSAZIONI remote

### AS400

#### Miglioramenti

- Segno di spunta vuoto rimosso per l'indicatore del livello di controllo
- Aggiunto il supporto per il nome esterno per le parole chiave IMPORT/EXPORT
- È stato aggiunto il supporto per %LEN sui campi
- CL - Aggiunto il supporto per i nuovi operatori per il linguaggio CLLE
- CL - Aggiunto il supporto per IF annidato
- COBOL - Migliore gestione del comando START quando utilizzato con più tasti
- DSPF - Migliore gestione della posizione del cursore con numero record

- DSPF - Migliorata la formattazione per i campi numerici con segno, solo numerici e per i campi su larga scala
- DSPF - Migliorata la determinazione del titolo per Screen General Help
- DSPF - Supporto migliorato delle specifiche di input/output
- DSPF - Migliore gestione dei separatori di raggruppamento durante la convalida del campo numerico
- Uscita di mappatura/record DDS migliorata
- Migliore capacità delle parole chiave REFFLT del file di stampa di risolvere i campi referenziati
- RPG - Supporto migliorato per le istruzioni «ALL free»
- RPG - Analisi delle condizioni migliorata e supporto aggiunto per la gestione di CABXX senza tag di risultato
- RPG - Migliore gestione delle specifiche di input dei campi numerici
- RPG - Migliore gestione delle chiamate di procedura all'interno delle condizioni IF/ELSEIF/WHEN
- RPG - Migliore gestione del comando READ quando viene chiamato su un file dspf
- RPG - Migliora il supporto per i file che si riferiscono a un DDS inesistente
- Migliora la gestione di REFFLD quando viene assegnato un nome in formato di registrazione fisico
- Aggiunto il supporto per l'utilizzo di «return» come nome di colonna db

## Funzionalità trasversali

### Nuove funzionalità

- Oracle: ha reso possibile definire gli utenti rispetto a SYS per archiviare le funzioni integrate

### Miglioramenti

- Versione Java aggiornata dalla v8 alla v17
- Condizione SQL migliorata con il nome della colonna Cluster
- È stato aggiunto il supporto per le clausole ORDER BY dalla visualizzazione

## Note di rilascio 3.9.0

Questa versione di AWS Blu Age Runtime and Modernization Tools si concentra su diversi miglioramenti trasversali del prodotto, volti ad aumentare le prestazioni nelle architetture ad alta

disponibilità, insieme a nuove funzionalità per portare l'esecuzione dei lavori a un livello superiore. Alcune caratteristiche e modifiche chiave di questa versione sono:

- Aggiornamento della versione da Angular 13 ad Angular 16, che aumenta la sicurezza e dà accesso a nuove funzionalità che migliorano le prestazioni delle applicazioni online dei clienti.
- Aggiungete il supporto delle funzionalità cross-job di AS400, con la caratteristica principale che le aziende possono inviare messaggi di richiesta in modo sincrono tra loro, permettendo il disaccoppiamento nei lavori modernizzati.
- Miglioramenti delle prestazioni sull'uso di Redis, tra cui l'ottimizzazione del pool di connessioni, l'elevata sicurezza della connessione e il meccanismo di blocco del set di dati aggiornato.

Per ulteriori informazioni sulle modifiche incluse in questa versione, consulta le seguenti sezioni.

## Versione di runtime 3.9.0

### ZoS

#### Nuove funzionalità

- Programma di ordinamento: ingressi VSAM aggiornati con lunghezza fissa
- JHDB DB: aggiunto un timeout configurabile

#### Miglioramenti

- Supporto migliorato per il separatore di riga per lo streaming se utilizzato nella concatenazione di file
- Supporto migliorato per l'apertura di file sequenziali concatenati. Inizializza dopo l'apertura DataSetIndex del file
- Supporto migliorato per il separatore decimale virtuale quando a NumericEditedType è influenzato da un valore numerico
- Supporto migliorato per valori non negativi NumericEditedType
- IDCAMS: le schede SYSIN vengono ora lette utilizzando la proprietà «encoding» definita in .yaml application-utility-pgm
- IDCAMS: grammatica aggiornata per supportare l'argomento FILE (..) nell'istruzione DEFINE CLUSTER

- INFUTILB: è stato aggiunto il supporto per l'argomento DFSIGDCB per sovrascrivere i parametri DCB di DD SYSREC
- INFUTIL: supporto migliorato per il parametro «DFSIGDCB YES»
- SPLICE migliorato per gestire file di input di grandi dimensioni
- DFSORT: gestione migliorata dei campi di commento
- DFSORT: aggiunto il supporto per il formato numerico in formato libero (firmato/non firmato) (SFF/UFF)
- SORT: è stato aggiunto il supporto di analisi per le istruzioni OPTION PRINT e OPTION ROUTE
- SORT/ICEMAN: aggiunto il supporto per le operazioni di divisione chiuse (campo con operatore DIV)
- Supporto migliorato per CICS READ utilizzando una chiave generica
- Funzione StringUtils .chargraphic fissa per rimuovere SOSI da un tipo di grafica
- Migliora le prestazioni su. DataUtils isDoubleByteCodifica
- JCL: supporto migliorato per la modalità di disposizione KEEP per un set di dati temporaneo. Il sistema cambia la disposizione in PASS
- JCL: gestisce i parametri DCB in modo dinamico
- JCL: uscite SUM FIELDS migliorate per valori errati
- JCL: CommonDDUtils: :getContent ora cerca RecordSize nel catalogo
- JCL: legge gli attributi RDW/RecordSize dal catalogo durante la creazione del set di dati
- JCL: Aggiunto il supporto per DCB=.MYDD per copiare i parametri DCB di un DD in un altro nella stessa fase di lavoro
- JCL: sistema di ereditarietà delle dimensioni dei record migliorato
- JCL: aggiunto il blocco esclusivo del set di dati (Redis)
- Redis: aggiunto il supporto SSL per la modalità standalone
- Redis: aggiunto il numero di blocchi Redis sincronizzato con blocco
- Redis: parametri Pool supportati per Redis lock
- Redis: aggiornamento ottimizzato dei metadati con Redis
- Redis: supporto migliorato per i cluster redis
- Miglioramento delle serrature aperte con modalità IO
- I set di dati migliorati bloccano le prestazioni e cancellano i blocchi non utilizzati

- Percorso migliorato del set di dati durante l'annullamento della registrazione del file
- Migliore invalidazione della cache della finestra di pre-fetch
- È stato aggiunto il supporto per l'utilizzo del provider di sorgenti dati di utilità threadsafe
- Controllo avanzato della nullità di DataSetState
- Supporto migliorato per non riaprire set di dati già aperti
- Maggiore robustezza per l'operazione finale del lavoro
- Supporto migliorato per gli indici, l'ordine delle chiavi, che consente la duplicazione
- Supporto migliorato per l'ordine di serializzazione degli skip list
- È stato aggiunto il supporto per la funzionalità di debug dump per aiutare a diagnosticare i problemi relativi all'ordine degli indici
- Supporto migliorato per l'aggiornamento dei metadati
- Supporto migliorato per la lettura in blocco di Blusam

## AS400

### Nuove funzionalità

- Crea un registro del contesto dell'applicazione
- Supporto per la parola chiave DSPF CLRL (NO) Supporta il monitoraggio dei blocchi dei record
- Support per keyed DataQueue
- Support per i messaggi INQUIRY per i lavori in batch
- È stato aggiunto il supporto per il file di stampa descritto dal programma per AS400 COBOL
- Gestisce il comando cl RMVJOBSCDE
- Miglioramento per RUNSQL/DLYJOB
- CHKOBJ: generazione del codice di errore precedente per il parametro LIB
- SNDPGMMMSG: supporta i parametri di stringa
- RTVDTAARA: sottostringa migliorata in LDA
- DSPFD: parametro FILE supportato aggiunto per un nome di file specifico
- RUNQRY: Supporto per file sql in QRY PARAM
- CRTDUPOB: Supporto per copiare i dati tra aree dati
- SBMJOB: converte le istruzioni in uso JobQueueManager

- OPNQRYF: Aggiunto il supporto per la libreria Qtemp
- CRTDUPOBJ: Logica migliorata per la copia del contenuto delle partizioni
- CRTDUPOBJ: Aggiunto il supporto per Qtemp per le visualizzazioni
- RTVSYSVAL: Supporto per il valore SYSVAL, QDATFMT nel comando CL
- CHKOBJ: Aggiunto il supporto per OUTQ
- RTVJOBA: supporta il parametro SWS
- SNDPGMMMSG e RCVMSG: sono supportati parametri aggiuntivi MSGF, MSGFLIB, MSGDTA, MSGTYPE, KEYVAR, MSGKEY, MSGID

## Miglioramenti

- Supporti migliorati per le schede I/O WORKSTATION
- Migliore gestione del messaggio impostato che si sovrappone al messaggio precedente
- Supporta informazioni aggiuntive sui messaggi su array-messageline
- Accesso agli array wrapper standalone migliorato all'interno di EVAL, SorTA, figurativi
- Migliora la pulizia delle DAO al termine dell'applicazione online
- Aggiunto il supporto per formati di data aggiuntivi e una migliore gestione degli input di stringhe
- Migliore gestione CVTDAT di SYSVAL aggiungendo la classe helper per i valori di sistema. Decodifica e costruisci i parametri dal comando CL SbmJob
- Pacchetto gapwalk-cl-command com.netfective.bluage.gapwalk.rt.blu4iv dalla scansione dei componenti
- È stato migliorato il supporto dei messaggi predefiniti per l'API di coda dei messaggi
- È stato migliorato il supporto retrieveSubfileRecord per la registrazione scritta in un altro programma
- È stato migliorato il supporto dei messaggi immediati per l'API di coda dei messaggi
- Migliore gestione dell'area dati locale durante l'invio di un lavoro
- Si avvia JobQueues automaticamente all'avvio del server
- Utilizza la configurazione ApplicationContext per decodificare i parametri per SBMJOB
- Miglioramento dei messaggi di errore forniti dal sistema
- Consente a RTVMSG di cercare i file.properties nelle sottodirectory annidate
- Gestisce il ripristino delle entità associate a puntatori errati/non validi

- Migliorato MessageHandlingBuilder per visualizzare msgld e name come stringhe per RCVMSG MsgFile
- Metodo di denominazione migliorato dell'API withMsgFile di accodamento dei messaggi
- Meccanismo di blocco dell'area dati migliorato
- RTVMBRD: Supporto per lettere minuscole e maiuscole per il parametro FILE
- CRTDUPOBJ: Migliore gestione delle viste
- CPYTOSTMF: migliore gestione della connessione
- CPYF: miglioramento nella gestione del nome della directory durante la copia da un file flat
- RCVF: gestisce correttamente i parametri DEV/RCDFMT e la trasformazione di RCDFMT per groovy e java
- RCVF: gestisce le chiamate successive ed evita di resettare il cursore
- CPYF: Aggiunto il supporto per la scrittura da file flat
- CRTDUPOBJ: aggiunta la gestione del nuovo obj con la libreria Qtemp
- CHGDTAARA: lunghezza massima dell'area dati aumentata da 256 a 2000
- AVOBJ: Assicurati che i record salvati siano in ordine di inserimento
- RTVDTAARA: Valori recuperati (non da tagliare)
- CHKOBJ: restituisce i messaggi di monitoraggio corretti quando il membro non esiste
- RTVDTAARA: Aggiunto il supporto della sottostringa LDA
- RTVDTAARA: restituisce spazi bianchi fino alla lunghezza della variabile specificata nel parametro RTNVAR
- RTVDTAARA: supporta i parametri interi per l'inizio e la lunghezza e supporta il formato di trasformazione più recente
- CHGDTAARA: è stato aggiunto il supporto per i parametri che includono i limiti inferiore e superiore
- CHKOBJ: gestisce il valore VIEW per il tipo di oggetto del parametro
- CHKOBJ: risultato impostato su true indipendentemente dal membro, se la vista esiste

## Capacità trasversali

### Nuove funzionalità

- Gestisce la generazione di report su file.txt
- Aggiunta la proprietà di origine dati CurrentSchema XA al gestore segreto

- Aggiungi la proprietà `yml.database.cursor.raise.already.opened.error` per consentire al framework di generare l'errore `SQLCODE 502` quando il cursore già aperto si apre

## Miglioramenti

- Aggiunte poesie `gapwalk` a AWS Blu Age sulla confezione Amazon EC2
- Utilizza il nuovo paradigma del gestore di segnali per impostazione predefinita
- Aggiunge il supporto per il blocco quando la disposizione è `MOD` o `OLD`
- Aggiunta cache per memorizzare i modelli di data e ora del database
- Funzione di controllo migliorata di `PackedType`
- Migliora le funzioni `DataUtils .setTo` per `Records` con `VariableSizeArray`
- Gestisce l'opzione `MQ SYNCPOINT` per quanto riguarda l'unità di esecuzione
- Framework abilitato per impostare `SQLCODE` sulla transazione di `rollback`
- Aggiunto il nome automatico della classe del driver in base al segreto della chiave del motore
- Timeout del programma/transazione
- Ripristina la posizione del cursore dopo il `rollback` quando si accede al cursore

## Terza parte

- Aggiorna `SnakeYAML`, `Redisson` e `Amazon SDK`, rimuovi `YamlBeans` (mitigare `CVE-2022-25857`, `CVE-2023-24621`, `CVE-2023-42809`, `CVE-2023-44487`)

## Versione 3.9.0 degli strumenti di modernizzazione

### ZoS

#### Miglioramenti

- Supporto migliorato per `XML-TEXT` come sorgente per la destinazione di tipo `String`
- Flusso di lavoro da `STM` a `UML` migliorato per supportare il modello di divisione `X/ (Y/Z)`
- `JHDB DB`: accetta la chiamata `ROLLBACK` prima di qualsiasi aggiornamento del database
- `JHDB DB`: accetta `ROLLBACK` anche se la transazione viene terminata (`NOP`)
- `JCL`: funzione di convalida dei passaggi migliorata



- SORT: gestisce la funzione SUM con valori decimali negativi di zona
- COBOL: aggiunge il supporto per l'escape tra virgolette singole/doppie nelle stringhe letterali

## AS400

### Miglioramenti

- Migliorata la gestione del codice di modifica X tramite la funzione incorporata %editc aggiungendo zeri iniziali
- Migliore gestione dei soli campi di input (valore iniziale)
- Aggiunti tasti di azione per facilitare i dialoghi
- Record a piè di pagina della tabella dinamica che appare in basso
- Comando START gestito senza KEY PHASE per i file che specificano una RECORD-KEY effettiva
- È stato aggiunto un valore predefinito per i tipi float e: :pow NumberUtils
- È stato aggiunto il supporto per la definizione di una variabile utilizzando LIKE (IN)
- Gestione del ciclo FOR aggiornata per supportare l'omissione di elementi opzionali
- Analisi RPG aggiornata per associare i record al nome dell'array CTDATA
- Migliore gestione degli indicatori per le dichiarazioni CABxx
- Supporta il parametro opzionale sulla parola chiave COMMIT
- Supporto migliorato per le parole chiave FORMAT in LF
- Codice operativo LOOKUP gestito con indicatori alto e uguale (o basso e uguale)
- Nome della chiave PF gestita dichiarato tra virgolette
- È stata migliorata la gestione di EDTCDE X per non sopprimere gli zeri iniziali
- Supporto migliorato per MSGCON nei file di stampa che non generano etichette senza nome
- Il campo CONTENT è condiviso da più strutture di dati
- Parametro ERRSFL gestito in combinazione con SFLMSG/SFLMSGID
- Codice principale migliorato prima dell'ambito di dichiarazione di Proc di Full Free RPG
- Aggiunta la specifica del controllo condizionato di analisi
- Supporto migliorato per il metodo setErrSfl () in dataholdermapper
- Risoluzione dei tipi migliorata per le variabili create internamente
- Supporto migliorato per l'opcode Z-ADD

- Migliorata la gestione del campo costante con valore DFT
- Migliora il supporto del campo intero all'interno degli annunci di stato del programma
- Ha gestito l'assegnazione degli indicatori nei parametri ENTRY
- Migliorato il filtro delle parole chiave propagate tramite la parola chiave ref/reffield
- Struttura DataArea dati senza nome supportata
- Migliore gestione del tipo di dati del puntatore
- Gli elementi gestiti dell'array utilizzati per definire le variabili con la parola chiave LIKE supportano l'accesso all'array nel campo di output
- Supporto migliorato per i caratteri numerici firmati, con visualizzazione solo di cifre
- Relazione logica supportata sulla scheda O
- Caso di test per %CHAR su formato alfanumerico
- Parola chiave di specifica di controllo supportata main
- EDTCDE con due parametri nel file della stampante
- Analisi RPG migliorata FullFree
- È stata migliorata la tabella dinamica per garantire che il piè di pagina sia posizionato correttamente
- Aggiunto il supporto per l'inizializzazione dei tipi numerici con la costante figurativa ALL
- Migliore gestione di più file logici RPG che fanno riferimento allo stesso file fisico
- Migliora il rilevamento dei campi modificati in una schermata moderna
- Sincronizzazione modale con campi dinamici
- Migliorata la gestione dei soli campi numerici firmati in uscita
- Migliora il supporto delle schede I/O WORKSTATION

## Funzionalità trasversali

### Nuove funzionalità

- Strumento Data Migrator: aggiunta la proprietà ebcDicFilesWith VarcharIn VB per consentire di tenere conto della lunghezza di 2 byte di VARCHAR durante la lettura dei byte
- Implementata un'API comune per registrare gli errori
- Implementazione BluAgeErrorDictionaryUtils e utilizzo di un'API comune per registrare errori e/o informazioni in COBOL2Model, RPGCycleBuilder, Definitions2Model e FieldsProcessor

- Grammatica SQL migliorata per supportare diverse definizioni di clausole di isolamento

## Miglioramenti

- Versione Angular aggiornata alla v16
- Angular: versione ajv aggiornata da 6 a 8.9

## Terza parte

- Groovy aggiornato alla versione 2.4.15

## Note di rilascio 3.8.0

Questa versione di AWS Blu Age Runtime and Modernization Tools si concentra su diversi miglioramenti trasversali del prodotto per migliorarne la qualità e la sicurezza, oltre a miglioramenti delle prestazioni per la memorizzazione nella cache e l'unificazione dei comandi supportati in un'unica distribuzione. Alcune funzionalità e modifiche chiave di questa versione sono:

- Aggiornamento della versione da Spring 2.5 a Spring 2.7, che aumenta il supporto di manutenzione, le prestazioni e la sicurezza della piattaforma.
- Unificazione di oltre 82 comandi CL supportati come parte della distribuzione per facilitare l'utilizzo e la over-the-counter distribuzione di applicazioni modernizzate che in precedenza utilizzavano lo scripting CL.
- Nuove API disponibili per operare e interagire meglio con i set di dati BluSAM, come l'importazione integrata nel servizio gestito e la capacità di elencare le informazioni sui metadati dei set di dati.
- Miglioramenti delle prestazioni ed estensione dell'utilizzo di Redis, inclusa la disponibilità in modalità cluster, il recupero dei dati ad alta disponibilità, la standardizzazione dell'uso dei segreti.

Per ulteriori informazioni sulle modifiche incluse in questa versione, consulta le seguenti sezioni.

## Versione di runtime 3.8.0

### ZoS

#### Nuove funzionalità

- Gestione della definizione della chiave come stringa per DynamicFileBuilder

- DFSORT: è stato aggiunto il supporto per più elementi nell'inizializzazione della grammatica OUTFIL TRAILER1 + DFSORT
- Strumento CommonDDUtils: gestione delle dimensioni dei record nei dati in-stream
- File indicizzato: gestione dell'opzione GENKEY

## Miglioramenti

- Servizi di caricamento BluSAM esternalizzati in un jar separato
- Aggiunto il supporto per configurare la posizione per l'archiviazione dei file temporanei
- Meccanismi di cache condivisa migliorati per i casi con più nodi
- Utilizzo della cache condivisa: IDCAMS verifica l'ottimizzazione
- Migliora l'iniezione ROWID per la selezione integrata
- JCL: Ogni procedura di lavoro in-stream viene ora generata in un file Groovy distinto
- Garantisci una copertura del 2% card-demo-v sulle schede IDCAMS JCL
- BluSAM: evita il riscaldamento duplicato quando si utilizzano più istanze
- Riduzione dell'ingombro di memoria sull'idratazione della cache
- Supporto per la configurazione del pool Jedis
- Aggiunto un separatore di riga allo streaming se utilizzato nella concatenazione di file
- Support per schede EBCDIC + commenti a blocchi (/.../) nell'utilità IDCAMS
- Interrogazione di supporto al database: supporto per stringhe a doppio byte nella conversione di level49 verso SQL
- Grammatica DFSORT: implementa 17 istruzioni di controllo + integrazione di 2 di esse (OMIT/INCLUDE)
- Migliora le colonne GRAPHIC e recupera INFUTILB
- Support per la lettura di file con tabella di dimensioni variabili
- Support per ZonedType con nibble signed dove il primo bit dell'ultimo byte è 'E'
- DFSORT/ICETOOL aggiunge il supporto per l'argomento NOMATCH =(..) se un record non corrisponde a nessuna delle costanti di ricerca CHANGE
- Compatibilità con Redis Cluster
- Gestione dello stato del lavoro (non riuscita) in base al codice di uscita groovy

- Supporto CICS SYNCPOINT ROLLBACK migliorato.
- Finestra di pre-fetch per ottimizzare l'utilizzo della cache Redis
- JCL/GROOVY: eredita la proprietà isRDW dal set di dati del passaggio precedente quando DISP=(, PASS)
- Gestione della copia parziale dei dati con un array di dimensioni variabili

## AS400

### Nuove funzionalità

- Support per schede I/O per file di visualizzazione
- Support per informazioni aggiuntive sui messaggi per le parole chiave DSPF ERRMSGID e CHKMSGID
- Support per più messaggi di errore sullo schermo frontend
- Aggiunto o migliorato il supporto di 82 comandi CL all'interno dell'applicazione gapwalk-cl-command

### Miglioramenti

- Supporto migliorato per DELETE e READ sotto il controllo degli impegni
- ConvertDate all'interno del file %dec integrato
- Intestazioni di sicurezza XSS applicate
- Maggiore robustezza e coerenza della generazione STM (migliore gestione di: riga di continuazione in formato libero rpg, virgole per la parte decimale, blocchi in formato libero nella definizione/dichiarazione)
- Generazione DataHolderMapper migliorata
- Maggiore robustezza e portata di modifica in DataAreaFactory
- È stato migliorato lo spostamento dell'attenzione sul tasto tab
- Prestazioni migliorate nella generazione di report Jasper
- Visualizzazione decimale migliorata con imbottitura 0s
- Supporto migliorato per il campo ROW/COL in INFDS
- Migliora il supporto per i campi modificati dalla schermata

- Aggiunti getter per il nome e il percorso del report generato
- Migliorata la lunghezza della coda dati
- Configurazione automatica migliorata di Job Queues per soddisfare i nuovi standard in Spring Boot 2.7
- Aggiornamenti migliorati delle workstation per più sessioni simultanee

## Funzionalità trasversali

### Nuove funzionalità

- Support per No Invalid Data Tolerance for Packed
- Aggiunta la paginazione/filtraggio per elencare gli endpoint del set di dati

### Miglioramenti

- Strategia di trasformazione delle query ORACLE migliorata nel confronto tra colonne e stringhe vuote
- Gestione di BLOB DB2 con i programmi di utilità DSNTEP e INFUTILB. I BLOB DB2 sono ora modernizzati in postgres di tipo BYTEA.
- Miglioramento della cancellazione dell'ultimo elemento del cursore
- Supporto migliorato per l'eliminazione del file RRDS
- Prestazioni segrete AWS Blusam migliorate
- Migliore gestione delle connessioni al database nel framework SQL
- Chiavi di gestione segrete standardizzate AWS con più origini dati
- Correzioni alla regressione delle prestazioni
- Funzione di controllo migliorata per PackedType
- Migliore gestione di LOW-VALUE per PackedType
- Pacchetto Spring Security aggiornato per Cognito Connection
- Non applica la codifica e la decodifica codeshiftpoint sui database di destinazione DB2

### Terza parte

- Aggiornamento Spring Boot da 2.5 a 2.7

## Versione 3.8.0 degli strumenti di modernizzazione

### ZoS

#### Nuove funzionalità

- JCL: gestione del flusso con carriage return «\ r»

#### Miglioramenti

- Registrazione migliorata per evitare la divisione per zero durante la modernizzazione di una clausola DIVIDE con ON SIZE ERROR
- JCL: supporto migliorato per la chiamata di una procedura in una procedura
- Supporto per la parola chiave OF nel comando FORMATTIME CICS in presenza di campi ambigui
- JCL: supporto per il carattere ¥ nelle variabili
- JCL: calcolo di RC in base ai passaggi precedenti
- Confronto tra byte anziché stringhe quando si utilizza PL1 SUBSTR
- Miglioramento dell'inizializzazione di array multidimensionali da un'unica fonte
- Migliore analisi di COBOL quando si tratta di una singola query SQL in un blocco IF

### AS400

#### Nuove funzionalità

- Supporto per l'istruzione IF annidata in CL
- Supporto migliorato per l'istruzione ENDDO in formato libero RPG

#### Miglioramenti

- Supporto migliorato per il livello di controllo del condizionamento
- Migliore restituzione dei prototipi con LIKE
- Supporto migliorato per la gestione delle funzioni %mesi, %anno, %giorni
- Supporto per la funzione di aiuto per l'intero schermo
- Gestione degli SPAZI VUOTI figurativi passati come parametro

- Miglioramento dell'espressione EVAL con l'operatore «»
- Gestione del comando START senza KEY PHASE
- Miglioramento nella gestione della parola chiave LIKERECD
- Miglioramento dei sottocampi senza nome
- Miglioramento della procedura di restituzione di un tipo senza segno
- Supporto migliorato per l'operazione RESET (Free RPG), integrazioni %CHAR e %DEC
- Miglioramento della funzione integrata %LOOKUPXX
- Supporto migliorato per la parola chiave LIKEDS sulla procedura senza prototipo
- Gestione del tipo di array di parole chiave Dim (VAR, AUTO)
- Supporto migliorato per XFOOT
- COBOL: supporto migliorato per i campi RENAMES
- CL: supporta la condizione while (true)
- Migliorata la gestione degli array autonomi con la parola chiave LIKE
- Miglioramento della funzione integrata %INT
- Analisi RPG Full Free migliorata
- Supporto migliorato per l'array nel collegamento
- CL2GROOVY: Dichiarazione Support Select
- Miglioramento della parola chiave DSPF «ERRMSGID»
- Migliorata la gestione dell'inizializzazione dei byte con zeri iniziali
- Miglioramento dei valori autorizzati per i campi numerici
- Gestione dell'extender H per l'istruzione EVAL in formato libero
- Da CL a Groovy: sottostringa Support di LDA
- Supporto migliorato per RESET su un record
- Migliorata la gestione di EDTCDE ed EDTWRD con riferimenti
- Migliore mappatura dei campi di input con campi DDS
- Supporto migliorato per lo spostamento di un carattere nell'array IN
- Miglioramento del prototipo con la parola chiave LIKEDS
- Supporto migliorato per la parola chiave DSPF DSPF DSPATR



- Analisi migliorata della scheda D con +/-
- Maggiore robustezza nelle chiamate ai programmi
- Maggiore robustezza nel processo di risoluzione sul campo

## Capacità trasversali

### Miglioramenti

- FrontEnd: simula l'evento paste per l'input IME

### Terza parte

- Aggiornamento Spring Boot da 2.5 a 2.7

## Note di rilascio 3.7.0

Questa versione di AWS Blu Age Runtime and Modernization Tools include principalmente miglioramenti per supportare meglio comandi e utilità, funzionalità di integrazione con AWS Secrets Manager e nuove funzionalità di monitoraggio. Alcune delle modifiche principali di questa versione sono:

- Più componenti di runtime possono ora utilizzare AWS Secrets Manager per aumentare la configurazione di sicurezza delle applicazioni modernizzate, principalmente relative alle fonti di dati delle utilità, Redis for TS Queues, BluSam cache e blocchi.
- Endpoint di monitoraggio che consente di recuperare i parametri di transazione, batch e JVM per l'ottimizzazione dell'utilizzo delle risorse e la gestione operativa, come stato, durata, volume e altro.
- Nuove funzionalità per supportare le chiamate IBM MQ in RPG e una maggiore copertura di trasformazione JCL SORT e IDCAMS.

Per ulteriori informazioni sulle modifiche incluse in questa versione, consulta le seguenti sezioni.

## Versione di runtime 3.7.0

### Argomenti

- [zOS](#)
- [AS400](#)

- [Capacità trasversali](#)

## zOS

### Nuove funzionalità

- Migliora l'analisi delle query coinvolte nell'applicazione di utilità del programma utilizzando la grammatica simile a SQL. (V7-9401)
- Gestisce un array di dimensioni variabili indicizzato durante l'offset (V7-9904)
- Supporta la colonna INSERT SQL TIME in DB2 con formato 24:00:00 ora (V7-10023)
- Supporta la query INSERT SQL da array con opzioni FOR ROWS e ATOMIC (V7-10105)
- JCL SORT: migliora per supportare OUTREC con IFTHEN ( TranscodeTool V7-10124)
- JCL SORT: aggiungi il supporto per la parola chiave DATE nel comando OUTREC (V7-10125)
- JCL: aggiunge il supporto per le procedure In-Stream (V7-10223)

### Miglioramenti

- Un set di dati contrassegnato con la disposizione «PASS» dovrebbe essere disponibile in tutte le fasi del lavoro (V7-9504)
- Supporta l'attributo JCL SCHENV (V7-9570)
- Supporta SEND con opzione CTLCHAR (V7-9714)
- COBOL - Gestisce diversi set di caratteri separatori di riga nelle istruzioni ACCEPT (V7-9875)
- Evita il rollback multiplo (V7-9958)
- Consenti l'uso della disposizione MOD da aggiungere alla fine dei file GDG (V7-10031)
- Ottimizzazione: refactoring PutAll (V7-10063)
- PutAll refactoring: aggiunta della paginazione (V7-10063)
- Rendi configurabile il timeout di lettura del client Jedis (V7-10063)
- UseSsl supporto per la modalità standalone (V7-10114)
- Supporta EIBDS dopo aver aperto il file con successo (V7-10147)
- Supporta EIBDS dopo una richiesta di controllo dei file (V7-10147)
- Migliora il supporto CICS SYNCPOINT (V7-10187)

- BluesamRedisSerializer: problema con MetadataPersistence (V7-10202)
- Supporta Redis AWS Secrets Manager per le code TS (V7-10204)
- Supporta JCLBCICS sulla personalizzazione della dimensione del nome DD (V7-10224)
- Aggiunge il supporto per il percorso assoluto nell'istruzione IDCAMS DELETE (V7-10308)

## AS400

### Nuove funzionalità

- Implementazione della funzione di aiuto per gli schermi AS400 (V7-9673)

### Miglioramenti

- Numero di record in INFDS (V7-9377)

## Capacità trasversali

### Nuove funzionalità

- Support for Runtime on EC2 per l'invio di log ad Amazon CloudWatch (D87990246)
- È stato aggiunto un nuovo endpoint per recuperare le metriche relative a batch, transazioni e JVM (D88393832)

### Miglioramenti

- Supporta le origini dati AWS Secrets Manager per l'utilità pgm (V7-9570)
- Aggiunto il supporto Db2 per DSNUTILB DISCARD (V7-9798)
- Supporto per la scrittura nel logger anziché nel flusso di output di sistema predefinito nei file SYSPRINT e SYSPUNCH predefiniti (V7-10098)
- Supporta la cache BluSam Redis e blocca le proprietà di connessione in AWS Secrets Manager (V7-10238)
- Support per la connessione SSL su Db2 XA AWS secret (V7-10258)
- Metadati aggiornati per IDCAMS REPRO e VERIFY (V7-10281)
- Migliore gestione del codice di restituzione IDCAMS Abend (V7-10307)

## Versione 3.7.0 degli strumenti di modernizzazione

### Argomenti

- [ZoS](#)
- [AS400](#)
- [Funzionalità trasversali](#)

### ZoS

#### Nuove funzionalità

- PLI - Assegnazione migliorata per gli array a sezione trasversale e bidimensionali (V7-9830)

### AS400

#### Nuove funzionalità

- Gestione degli indicatori del livello di controllo (V7-9227)
- Supporto per il parametro EXTNAME \*INPUT (V7-9897)
- Enhanced Goto Rewriting: Supporto per i tag che si trovano nelle istruzioni SELECT OTHER (V7-9973)
- Supporta la parola chiave REFRESH DSPF (V7-10049)

### Miglioramenti

- Miglioramento nella gestione della parola chiave di descrizione dei file EXTIND (\*Linux) (V7-7404)
- Migliore trasformazione dei file SQLDDS (V7-7687)
- Gli oggetti di file non sono più generati per i file AS400 (V7-9062)
- Migliore gestione della parola chiave di descrizione dei file EXTDESC (V7-9268)
- Migliore gestione del componente integrato %CHAR (V7-9311)
- Supporto migliorato per il pagedown sull'ultimo record senza SFLEND (V7-9322)
- Supporto migliorato per strutture di dati con prefisso (V7-9436)
- Supporto per la dimensione definita con %SIZE (V7-9472)

- Supporto per la gestione del nome di campo PF dichiarato tra virgolette (V7-9557)
- Funzionamento migliorato dei file - senza distinzione tra maiuscole e minuscole (V7-9785)
- Supporto per il campo inizializzato su\*USER (V7-9806)
- Supporto per il tipo COMP in AS400 (V7-9840)
- Analisi COBOL400 migliorata su (Not) (V7-9922) InvalidKey
- Migliore gestione delle operazioni SCAN (V7-9971)
- Supporto migliorato per l'opcode GOTO (V7-9973)
- Migliore gestione del funzionamento EXCEPT (V7-9977)
- Supporto migliorato per i prefissi (V7-10000)
- Supporto per chiamate MQ in RPG (V7-10007)
- %LOOKUP integrato migliorato (struttura dati ad array con chiavi) (V7-10022)
- Supporto per il funzionamento Close \*All (V7-10036)
- Supporto per l'istruzione SQLDDS UPDATE AS ROW CHANGE (V7-10051)
- Miglioramento della gestione del tipo di valore letterale Long (V7-10073)
- Grammatica RPG migliorata (uso della parola chiave INZ come nome della subroutine) (V7-10074)
- Grammatica RPG migliorata per supportare valori numerici con parte frazionaria vuota (V7-10077)
- Supporto migliorato per i campi condivisi tra CL e file esterno (V7-10081)
- Supporto migliorato per gli indicatori condizionali DDS (V7-10084)
- Supporto per il tipo binario DDS con programmi COBOL (V7-10100)
- Migliore collisione dei nomi con collegamento (V7-10109)
- Supporto per la combinazione delle procedure principali e di esportazione (V7-10112)
- Supporto migliorato DataStructure in una sottoprocedura (V7-10113)
- Supporto migliorato per CLEAR (V7-10126)
- Supporto migliorato del loop DO (V7-10134)
- Supporta SQLTYPE in un gioco di ruolo completamente gratuito (V7-10151)
- Migliore analisi delle condizioni sulla parola chiave DDS (V7-10155)
- Generazione DSL migliorata (V7-10163)
- Miglioramento per ProcessIndicators quando la condizione è un'espressione binaria. (V7-10164)

- GoTos con condizione Else migliorata (V7-10168)
- Support per i tipi Time e Timestamp in DSPF (V7-10173)
- Analisi migliorata della riga di continuazione per DDS (V7-10183)
- Supporto COBOL per RENAMES FLD OF RECORD (V7-10195)
- Migliore analisi condizionale degli indicatori sui campi DSPF (V7-10221)
- Supporta l'analisi della parola chiave DDS NOALTSEQ (V7-10288)
- Menu Support Help e campi nascosti (V7-10314)
- Controllo dell'integrità delle parole chiave di aiuto DSPF migliorato (V7-10328)
- Non si propagano più tutte le parole chiave nel campo Ref (V7-10347)

## Funzionalità trasversali

### Nuove funzionalità

- Data Migrator: gestione dei dati CLOB (V7-9665)

### Miglioramenti

- Propagazione della proprietà JCL SCHENV da JOB alla definizione PROC GROOVY tramite (V7-10225) JobContext
- FrontEnd - Regolazione delle dimensioni della finestra in caso di assenza di bordi (V7-10358)

## Note di rilascio 3.6.0

Questa versione di AWS Blu Age Runtime and Modernization Tools offre nuove funzionalità per le migrazioni precedenti di zOS e AS400, principalmente orientate all'espansione dei meccanismi di supporto CICS, all'integrazione delle funzionalità JCL, all'ottimizzazione delle prestazioni nelle funzionalità simultanee e ad alto volume e all'aggiunta di funzionalità. multi-data-source Alcune delle modifiche chiave di questa versione sono:

- Miglioramento della gestione dinamica dei file JCL, espansione delle istruzioni correnti e gestione di set di dati concatenati, esecuzione di più istruzioni in un unico blocco e trasferimento dei dati dai batch ai programmi.
- Supporto migliorato di più comandi CICS, inclusa la richiesta di diversi tipi di risorse CICS.

- La capacità di disporre di database diversi quando si utilizza Blu Age Runtime Utilities, ideale per scenari in cui i dati aziendali sono distribuiti su più fonti.

Per ulteriori informazioni sulle modifiche incluse in questa versione, vedere le seguenti sezioni.

## Versione di runtime 3.6.0

### Argomenti

- [zOS](#)
- [AS400](#)
- [Funzionalità trasversali](#)

### zOS

#### Nuove funzionalità

- JCL - DynamicFileBuilder - Gestione avanzata degli handle di file (V7-9408)
- Conversione del formato migliorata su alcune funzioni SQL DB2 integrate quando si chiama l'utilità INFUTILB UNLOAD (V7-9554)
- Assegnazioni di array multidimensionali PLI migliorate (V7-9592)
- Gestione del reindirizzamento sysout al file (V7-9992)

#### Miglioramenti

- Aggiunta l'attivazione delle stored procedure per DB2 RDBMS (V7-9155)
- SORT gestisce la conversione in formato PDF (V7-9286)
- JCL/GROOVY - Migliora l'istruzione REPRO per supportare i set di dati DUMMY (V7-9424)
- Migliora il supporto CICS UNLOCK (V7-9606)
- Gestisce la dimensione del valore predefinito per Union (V7-9648)
- JCL/GROOVY gestiscono terminazioni/disposizioni diverse in set di dati concatenati (V7-9653)
- Rendi PageSize configurabile per i set di dati blusam (V7-9680)
- DSNUTIL: consente il caricamento delle 24:00:00 come TIME valido in DB2LUW (V7-9697)
- Supporta il confronto HIGH-VALUES (0xff) in NumberUtils .ne () NumberUtils /.eq () (V7-9731)

- JCL/GROOVY - supporta DO... Parole chiave THEN nelle clausole IDCAMS IF-THEN-ELSE per eseguire più istruzioni in un singolo blocco (V7-9750)
- Programma chiamato JHDB non valido esterno BatchRunner a JHDB (V7-9782)
- Supporta i caratteri di spazio bianco nella scheda di controllo SORT OUTFIL (V7-9808)
- Migliora il supporto CICS READ PREV (V7-9845)
- Migliora l'accesso simultaneo agli indici dei set di dati (V7-9864)
- Migliora il supporto CICS REWRITE (V7-9873)
- COBOL: supporto per SYSIN multilinea nelle istruzioni ACCEPT per passare i dati dal batch (JCL) a un programma (COBOL) (V7-9875)
- Groovy - Migliore gestione della fase di creazione dei file (V7-9876) ConcatenatedFileConfiguration
- IDCAMS UTILITY - Gestione dell'istruzione DEFINE PATH (V7-9878)
- SORT BUILD - Modifica l'opzione TRAN e gestisce gli spazi vuoti impliciti (V7-9925)
- Migliora CICS DELETE con il supporto dell'opzione GENERIC (V7-9939)
- Migliora il supporto CICS STARTBR ed ENDBR (V7-9952)
- Migliora le prestazioni di chiusura in caso di accesso simultaneo (V7-9953)
- Migliora la gestione dello stato dei file all'avvio (V7-9991)
- Groovy: consente la chiamata di `getDisposition ()/(/()` su `(getNormalTerminationV7-10012)` `getAbnormalTermination ConcatenatedFileConfiguration`

## AS400

### Nuove funzionalità

- Supporta indicatori esterni sulle parole chiave COMMIT (V7-6035)
- Reimposta il loop ReadC dopo la scrittura SFLCTL (V7-8061)
- Supporta l'indicatore LR in CALL (V7-9250)
- Aggiungi un nuovo tipo di campo dinamico (diviso) per gestire il campo di input su più righe (V7-9370)
- Supporta file primario/secondario (V7-9390)
- Le aree dati locali vengono ora passate al lavoro chiamato quando si invia un lavoro (V7-9775)
- Supporto di QTEMP per l'area dati e supporto per la creazione di valore dell'area dati. (V7-9916)



- Commitment Control: supporto per abilitare/disabilitare il controllo degli impegni (V7-9956)
- Supporta gli indicatori esterni sulle parole chiave COMMIT

## Miglioramenti

- Migliora la visualizzazione dei valori 0 e l'EDTWRD (V7-8933)
- Supporto della parola chiave DSPF «CHKMSGID» (V7-9125)
- Transazione di commit SQL al termine del batch (V7-9232)
- Migliora il supporto delle parole chiave EXPORT e IMPORT per campi e strutture dati (V7-9265)
- Supporta lettere minuscole DateHelper (V7-9461)
- Supporta la conversione da \*CYMD a \*ISO (numerico) (V7-9488)
- Migliora la gestione del %len integrato per un campo variabile (lato sinistro e destro di un'espressione) (V7-9733)
- Migliora il supporto per le funzioni integrate '%LOOKUPXX' XX («LE», "LT», "GE», "GT») (V7-10064)

## Funzionalità trasversali

### Nuove funzionalità

- CICS - Migliora la transazione Inquire per lo stato dell'opzione (V7-9712)
- JCL - Migliora il caricamento di sysprint con il file di output del sistema (V7-9797)
- CICS - Migliora INQUIRE TSQUEUE (V7-9823)
- CICS - Migliora il terminale Inquire per l'opzione userid (V7-9906)

### Miglioramenti

- Migliora la gestione del confronto con blank (V7-8047)
- Migliora la registrazione per Jics e (V7-8847) BluSam
- Supporta gli attributi estesi BMS SOSI e il simbolo programmato F8 per i campi dinamici (V7-8857)
- Gestisci l'overflow del buffer nei parametri del programma (V7-9138)
- Migliora la concorrenza di scrittura dei thread per il registro Blusam Locks (V7-9505)

- Supporta la configurazione di più origini dati per Utility-PGM (V7-9570)
- Modalità solo blocco del livello di record Blusam (V7-9626)
- Garantire che la persistenza dei metadati resista al riavvio del server (V7-9748)
- Migliora la pulizia DAO in caso di eccezione (chiusura del browser) (V7-9790)
- Supporto DummyFile per INFUTILB SYSPUNCH (V7-9799)
- Migliora il supporto per i valori negativi su (V7-9935) NumericEditedType

## Versione 3.6.0 degli strumenti di modernizzazione

### Argomenti

- [ZoS](#)
- [AS400](#)
- [Funzionalità trasversali](#)

### ZoS

#### Nuove funzionalità

- JCL - Migliora la registrazione per la fine della procedura (V7-8509)
- PL1 - Migliora la generazione di sacchetti per il tipo di dati (V7-8917) PakedLong
- JCL - Migliora la registrazione per la fine della procedura quando il file contiene il marker «end»// (V7-9509)
- PL1 - Migliora il supporto per GET EDIT con stream a virgola fissa e SYSIN (V7-9593)
- DB2 - Migliora il supporto per il tipo VARGRAPHIC DB2 (V7-9809)
- CICS - Migliora il comando QUERY SECURITY per l'opzione LOGMESSAGE (V7-9969)
- PL1 - Migliora la generazione di sacchetti per CHARG/Chargraphic built-in (V7-9989)

#### Miglioramenti

- PL1- Migliora il supporto per la parola chiave INCLUDEX (V7-9588)
- PL/I - Gestisce la parola chiave CHARGGRAPHIC come parametro valido per qualsiasi chiamata al metodo (V7-9589)

- Miglioramento della risoluzione variabile dell'host PL1 quando viene denominata con caratteri specifici @ # \$ §. (V7-9654)
- COBOL - Support delle parole chiave C01... C12 e S01... S05 come parametro dell'istruzione WRITE ADVANCING in fase di analisi (V7-9669)

## AS400

### Nuove funzionalità

- Supporta la trasformazione SQL-DDS in Analyzer (V7-7687)
- Automatizza il rilevamento dei file SQL-DDS (V7-7687)
- Implementazione della preelaborazione SQL-DDS (V7-7687)
- Parola chiave Support ALIGN (V7-9254)
- Support ExtName per DSPF e array multi-dim (V7-9663)
- InvalidKey Dichiarazioni di supporto su COBOL WRITE (V7-9793)

### Miglioramenti

- Miglioramento del codice operativo TESTB (V7-8865)
- Migliora il supporto di DECFMT on focus (V7-8933)
- Gestione dell'indicatore risultante su MOVE (V7-9224)
- Migliora il supporto della parola chiave TEMPLATE per campi e strutture dati (V7-9278)
- Miglioramento dei LIKEDS (il DS definito utilizzando LIKEDS viene qualificato automaticamente) (V7-9302)
- COBOL - Migliora la generazione della struttura degli indicatori (V7-9423)
- Il parametro Const nel prototipo non è di sola lettura (V7-9437)
- Migliora la parola chiave EDTCDE con il codice di modifica «Y» (V7-9443)
- Supporta la generazione del campo\*ROUTINE in PSDS e INFDS (V7-9487)
- Migliora la riscrittura del campo XXX in modalità standalone (il valore predefinito viene perso durante la riscrittura) (V7-9522)
- Migliora il supporto delle parole chiave DSPF (V7-9658)
- Gestione del valore predefinito ZEROES su binario (V7-9666)

- Supporta il puntatore implicito (V7-9719)
- Migliora la gestione della chiamata integrata %size con un solo parametro (V7-9730)
- Migliora la gestione dei riferimenti alla struttura dei dati nelle chiamate integrate (%ELEM) (V7-9736)
- Migliora la gestione della lunghezza firmata per i campi con riferimento LIKE nella specifica delle definizioni (V7-9738)
- Miglioramento rispetto a REWRITE (V7-9791)
- Miglioramento della generazione di indici a partire da file DDS (V7-9803)
- Migliora la robustezza dei mappatori con un valore numerico non valido (V7-9813)
- Migliora la generazione di file SQLModel e AllIndexes (V7-9818)
- Migliora il supporto DS qualificato (V7-9863)
- Migliora il supporto di LOOKUP (con un campo autonomo SIMILE a un DS nel parametro) (V7-9961)
- Migliora l'indicatore LIKE (V7-9985)
- Gestione dell'indicatore risultante su MVR (V7-9995)
- Supporta il carattere N con tilde (V7-10021)
- Migliora la generazione di file DDL moderni da file legacy SQLDDS (V7-10067)

## Funzionalità trasversali

### Nuove funzionalità

- Personalizza la posizione delle risorse con una proprietà yml (D88816105)
- COBOL - Supporto dell'istruzione EXIT PERFORM per uscire da un PERFORM in linea senza usare GO TO/PERFORM... TRAMITE (V7-9582)
- Specificare la codifica legacy predefinita da considerare nei metadati globali. (V7-9883)

### Miglioramenti

- Migliora la generazione di maschere (V7-9602)
- Migliora il riscaldamento del contesto (V7-9621)
- Rendi sicuro il thread Charset CUSTOM930. (V7-9674)

- Miglioramento rispetto a MOVEA (V7-9773)

## Note di rilascio 3.5.0

Questa versione di AWS Blu Age Runtime and Modernization Tools offre nuove funzionalità per le migrazioni precedenti di zOS e AS400, principalmente orientate all'ottimizzazione dei set di dati e della messaggistica, nonché funzionalità Java estese come risorsa risultante del processo di trasformazione. Alcune delle modifiche principali di questa versione sono:

- Capacità di migrare i programmi CL in Java in aggiunta alla funzionalità preesistente di groovy scripts, per facilitarne l'integrazione con altri programmi modernizzati e per semplificare la curva di apprendimento dei clienti unificando il linguaggio di programmazione risultante.
- Riduzione dei tempi e ottimizzazione delle prestazioni dei caricamenti dei set di dati in Redis con la nuova funzionalità Data Bulk.
- Capacità di gestire e trasmettere set di dati durante le fasi di lavoro per modernizzare i comportamenti dei set di dati tradizionali.
- Estensione della migrazione SQL per supportare i file di input VB e la migrazione semplificata di Java 11.
- Diversi nuovi meccanismi per un'integrazione più rapida con IBM MQ, tra cui intestazioni aggiuntive, supporto GET/PUT esteso e recupero automatico dei metadati della coda.
- REST Endpoint per set di dati, metadati e importazione di set di dati da bucket S3.

Per ulteriori informazioni sulle modifiche incluse in questa versione, consulta le seguenti sezioni.

## Versione di runtime 3.5.0

### Argomenti

- [ZoS](#)
- [COME 400](#)
- [Capacità trasversali](#)

## ZoS

### Nuove funzionalità

- JCL SORT - Gestisce la sovrapposizione di nuove parole chiave (V7-9409)
- ZOS COBOL: migliora il supporto dei caratteri mobili (V7-9404)
- Porta di TSQueue su & (V7-9212) RedisJics RedisTemplate ListOperations
- ZOS JCL: migliora il percorso della directory temporanea con la directory dei file se definita tramite (V7-9012) UserDefinedParameters
- Gestisci FUNCTION ORD-MAX con ALL (tutti gli elementi dell'array) (V7-9366)
- Le chiavi con prefisso e leggibili dall'uomo vengono ora utilizzate per archiviare le code TS in Redis (V7-9212)
- Aggiungi get dataset endpoint per l'API blusam
- JCL - AGGIUNGI il supporto per il lavoro in batch con nome che include caratteri speciali come # (V7-9136)
- Il recupero di TSMModel viene ora eseguito in modo affidabile su richiesta (V7-9212)

## Miglioramenti

- Supporto INCLUDE senza versione nei file LNK (V7-6022)
- MQ - Migliora il supporto per la codifica (V7-9652)
- Miglioramento del supporto per byte doppi o set di caratteri misti per tipi di caratteri diversi (V7-9596)
- JCL - Support della configurazione FilesDirectory nelle istruzioni IDCAMS delete NONVSAM (V7-9609)
- Supporta la modalità bulk per il caricamento di set di dati ESDS e RRDS da file (V7-8639)
- Gestisci l'apertura di ESDS vuoti in modalità di input. (V7-9287)
- Migliora l'istruzione DEFINE CLUSTER con il supporto per le abbreviazioni ORD/UNORD (V7-9451)
- BluSam Miglioramenti delle prestazioni di Redis Lock (V7-8639)
- Migliora l'istruzione DEFINE CLUSTER per supportare RECORDSIZE fornito nell'ambito degli argomenti DATA () (V7-9337)
- Aggiunge il supporto degli attributi BUFFERSPACE/UNIQUE sulle istruzioni DEFINE CLUSTER (V7-9419)
- Migliora l'operazione di BluSam lettura per set di dati di record a lunghezza variabile. (V7-9391)
- CICS ADDRESS rappresenta correttamente il CWA mancante come nullo (V7-9491)

- Rimuovi i blocchi di scrittura non necessari alla fine (V7-8639)
- Gestisci l'iniezione del modello di cache Redis nella cache (V7-9510)
- Decodifica correttamente il parametro BPXWDYN (V7-9417)
- Miglioramento del consumo di esportazioni LISTCAT (V7-9201)
- Supporto per caratteri non stampabili nel nome di TS Queues (V7-9212) BluSam
- Gestisci la creazione di mappe di ricezione per il campo con mapset null (V7-9486)
- Migliora le operazioni di BluesamRelativeFile cancellazione e riscrittura per la modalità di accesso dinamico. (V7-8989)

## COME 400

### Nuove funzionalità

- Aggiunge una funzionalità per generare file CL come programmi Java tramite pivot DS/STM standard (V7-9427)
- Supporta file di input con modalità ADD (V7-9378)
- Migliore gestione dell'ordinamento e del recupero per supportare il comando cl OPNQRYP (Open Query File) e aggiunto il supporto del parametro SHARE in. OverrideItem (V7-9364)

### Miglioramenti

- Supporta SFLNXTCHG su (V7-8061) UpdateSubfile
- Modifica l'ambito del contesto CL quando esegui il comando CL (V7-9624)
- Gestisci il codice di ritorno per il programma BPXWDYN (V7-9417)
- Monitor locali trasparenti. (V7-9624)
- Support della parola chiave DSPF RTNCSRLOC (V7-9389)
- setOnGreaterOrEqual() non impostato su Uguale a 1 (V7-9342)
- Aggiorna la cache dei campi su (V7-9376) UpdateSubfileRecord
- Improve Support SFLNXTCHG (V7-8061)

## Capacità trasversali

### Nuove funzionalità

- Ignora il prefisso G sulla stringa grafica letterale. (V7-9420)
- ZOS COBOL - Migliora il supporto di Fiedl.initialize () per alcune strutture speciali (V7-9485)
- Consenti l'inizializzazione del contesto in modo asincrono per migliorare le prestazioni di avvio del programma (V7-9446)
- SQL rilascia esplicitamente l'istruzione prepare aperta e. ResultSet (V7-9422)
- Enhance JMS MQ - supporto MQRFH2 per MQ PUT /V7-7085 - supporto del gestore di code predefinito (V7-9400)
- Gestione SQL: abilita le conversioni Lambda sui parametri per i comandi SET (V7-9492)
- ZOS MQ JMS - Aggiunge il supporto a MQCOMIT e MQBACK (V7-9399)
- ZOS IBMMQ - Migliora il supporto a MQINQ (V7-9544)
- Gestisci il funzionamento CONCAT con byte anziché con stringa quando usi la codifica a doppio byte. (V7-8932)
- ZOS IBMMQ - Migliora il supporto del comando PUT con le opzioni SET\_ALL\_CONTEXT (V7-9544)

## Miglioramenti

- Gestisci i nomi dei file gdg con il carattere \$ (V7-9066)
- SQL Diagnostic restituisce 1 come clausola NUMBER quando l'istruzione SQL precedente ha esito positivo. (V7-9410)
- Struttura per un campo con lunghezza non nulla (V7-7536)
- Supporta la funzione GRAFICA PL1 integrata (V7-9245)
- MQ - Aggiunto il supporto della versione per l'impostazione dei campi MQGMO (V7-9500)
- JMS MQ GET - Miglioramento della lunghezza dei dati restituiti dal messaggio (V7-9502)
- Imposta sqlerrd (3) con il numero di elementi recuperati nel contesto ROWSET. (V7-9371)

## Versione 3.5.0 degli strumenti di modernizzazione

### Argomenti

- [ZoS](#)
- [AS400](#)
- [Capacità trasversali](#)



## ZoS

### Nuove funzionalità

- ZOS PLI - Supporta l'indice asterisco nell'assegnazione con espressione binaria (V7-9178)
- Da JCL a BatchScript - A «//» segna la fine dell'esecuzione del lavoro (V7-9304)
- ZOS PLI: migliora il supporto di caratteri e segni mobili in formato numerico modificato (V7-8982)
- COBOL - Supporto della funzione SUM integrata (V7-9367)
- JCL- facoltativamente, commenta il codice morto dopo l'istruzione null (//) (V7-9202)
- JCL- Support dell'operatore '|' nella dichiarazione delle condizioni (V7-9499)
- PL/I - Commento delle direttive di precompilazione nella fase di preelaborazione per evitare l'analisi delle eccezioni (V7-9507)

### Miglioramenti

- Gestisci la definizione dello stream con delimitatore (V7-9615)
- Miglioramento della gestione delle esportazioni LISTCAT. (V7-9201)
- PL/I - Miglioramento per supportare argomenti «nulli» impliciti (V7-9204)

## AS400

### Nuove funzionalità

- Support della parola chiave DDS CONCAT (V7-9439)
- Rifattorizza il codice java generato per le parole chiave DSPF. (V7-7700)
- Support Variing keyword sui campi all'interno di una definizione della struttura dati (V7-9029)

### Miglioramenti

- Migliora l'analisi della relazione logica AND/OR (V7-9352)
- COBOL Migliora la mappatura tra vo e DSentity (V7-9449)
- Visualizza un valore vuoto se l'input numerico è focalizzato (V7-9374)
- Variabile locale in SQL Declare Cursor (V7-9456)
- Problema di ambito con DS vuoto (V7-9466)

- Tronca le righe dopo il col 80 prima dell'analisi (V7-9632)
- Migliora la gestione dei riferimenti ai campi e delle chiamate integrate nelle parole chiave (DIM, LIKE,...) nelle specifiche delle definizioni (V7-9358)
- Supporta commenti SQL (-- ) (V7-9632)
- FullFree analisi, digitare Date/Time/Timestamp (V7-9542)
- Includi FullFree SQLCA dall'analisi (V7-9333)
- Migliora il supporto del livello di controllo. (V7-9610)
- Gestisci il confronto DS con \*BLANKS (V7-9668)
- Migliora il supporto di più indicatori in DDS (V7-9318)
- Migliora il supporto di più programmi DSPF (V7-9657)
- Migliora la gestione del campo con LIKE (caso di struttura dati apprezzata e caso di struttura dati desiderata in un array) (V7-9213)
- RPG gratuito, gestisci la continuazione su Literal (V7-9686)
- Improve Support per i record di fine programma (V7-9452)
- Support della frase LINKAGE nell'istruzione CALL. (V7-9685)
- Codice operativo CASXX (CASBB senza gruppo CASXX) (V7-9357)
- FullFreeMigliora l'analisi dei giochi di ruolo (V7-9457)
- Il %LEN integrato non supporta DS come argomento (V7-9267)
- Miglioramenti di MOVEA quando il fattore 2 è \*ALL'X... ' (V7-9228)
- Supporta l'assegnazione con il campo RENAME (V7-9385)

## Capacità trasversali

### Nuove funzionalità

- Strumento SQL Migrator: aggiunge l'opzione OID per la lunghezza variabile dei record nella fase di caricamento di ebcdic. (V7-9380)
- Strumento SQL Migrator - Supporto per Java 11 su opzione OID (V7-9599)

### Miglioramenti

- Migliora il supporto per gli array annidati (V7-9595)

- Sostituisci `^` carattere con `!` nel caso in cui `^` sia supportato dalla codifica originale. (V7-9465)
- JCL - Support della terminazione normale PASS per condividere set di dati tra le fasi di lavoro (V7-9504)
- Applica ON NULL alla definizione di colonna su ORACLE quando si tratta di VARCHAR e del tipo di colonna db nullable. (V7-9681)
- Migliora la conformità all'iniezione a molla (V7-9635)

## AWS I concetti di Blu Age Runtime

Comprendere i concetti di base di AWS Blu Age Runtime può aiutarti a capire come le tue applicazioni vengono modernizzate con il refactoring automatizzato.

### Argomenti

- [AWS Architettura di alto livello Blu Age Runtime](#)
- [AWS Struttura Blu Age di un'applicazione modernizzata](#)
- [Semplificatore di dati](#)

## AWS Architettura di alto livello Blu Age Runtime

Come parte della soluzione AWS Blu Age per modernizzare i programmi legacy in Java, AWS Blu Age Runtime fornisce un punto di ingresso unificato e basato su REST per le applicazioni modernizzate e un framework di esecuzione per tali applicazioni, attraverso librerie che forniscono costrutti legacy e una standardizzazione dell'organizzazione del codice dei programmi.

Tali applicazioni modernizzate sono il risultato del processo AWS Blu Age Automated Refactor per modernizzare i programmi mainframe e midrange (indicati nel seguente documento come «legacy») verso un'architettura basata sul web.

Gli obiettivi di AWS Blu Age Runtime sono la riproduzione del comportamento (isofunzionalità) dei programmi legacy, le prestazioni (rispetto al tempo di esecuzione dei programmi e al consumo di risorse) e la facilità di manutenzione dei programmi modernizzati da parte degli sviluppatori Java, attraverso l'uso di ambienti e idiomi familiari come tomcat, Spring, getters/setters, API fluenti...

### Argomenti

- [AWS Componenti di runtime Blu Age](#)

- [Ambienti di esecuzione](#)
- [Apolidia e gestione delle sessioni](#)
- [Alta disponibilità e apolidia](#)

## AWS Componenti di runtime Blu Age

L'ambiente AWS Blu Age Runtime è composto da due tipi di componenti:

- Un insieme di librerie java (file jar) spesso denominate «la cartella condivisa» e che forniscono costrutti e istruzioni preesistenti.
- Un insieme di applicazioni Web (file di guerra) contenenti applicazioni Web basate su Spring che forniscono un set comune di framework e servizi per programmi modernizzati.

Le sezioni seguenti descrivono in dettaglio il ruolo di entrambi questi componenti.

### AWS Librerie Blu Age

Le librerie AWS Blu Age sono un insieme di file jar memorizzati in una `shared/` sottocartella aggiunta al classpath standard di tomcat, in modo da renderli disponibili a tutti i programmi Java modernizzati. Il loro obiettivo è fornire funzionalità che non siano né nativamente né facilmente disponibili nell'ambiente di programmazione Java, ma tipiche degli ambienti di sviluppo legacy. Queste funzionalità sono esposte in un modo il più familiare possibile agli sviluppatori Java (getters/ setter, API fluenti basate su classi). Un esempio importante è la libreria Data Simplifier, che fornisce costrutti di layout e manipolazione della memoria legacy (presenti nei linguaggi COBOL, PL1 o RPG) ai programmi Java. Questi jar sono una dipendenza fondamentale del codice Java modernizzato generato dai programmi legacy. Per ulteriori informazioni su Data Simplifier, vedere. [Semplificatore di dati](#)

### Applicazione Web

I Web Application Archives (WAR) sono un modo standard per distribuire codice e applicazioni sul server delle applicazioni tomcat. Quelli forniti come parte del runtime AWS Blu Age mirano a fornire una serie di framework di esecuzione che riproducono ambienti e monitor delle transazioni legacy (batch JCL, CICS, IMS...) e i servizi richiesti associati.

Il più importante è `gapwalk-application` (spesso abbreviato in «Gapwalk»), che fornisce un set unificato di punti di ingresso basati su REST per attivare e controllare l'esecuzione di transazioni, programmi e batch. Per ulteriori informazioni, consulta [AWS API Blu Age Runtime](#).

Questa applicazione Web alloca i thread e le risorse di esecuzione Java per eseguire programmi modernizzati nel contesto per il quale sono stati progettati. Esempi di tali ambienti riprodotti sono descritti in dettaglio nella sezione seguente.

Altre applicazioni Web aggiungono all'ambiente di esecuzione (più precisamente, al «Registro dei programmi» descritto di seguito) programmi che emulano quelli disponibili e richiamabili dai programmi legacy. Due categorie importanti sono:

- Emulazione di programmi forniti dal sistema operativo: i batch basati su JCL prevedono in particolare di poter richiamare una varietà di programmi di manipolazione di file e database come parte del loro ambiente standard. Gli esempi includono/o. SORT DFSORT IDCAMS A tal fine, vengono forniti programmi Java che riproducono tale comportamento e sono richiamabili utilizzando le stesse convenzioni di quelli precedenti.
- «Driver», che sono programmi specializzati forniti dal framework di esecuzione o dal middleware come punti di ingresso. Un esempio è CBLTDLI da quali programmi COBOL eseguiti nell'ambiente IMS dipendono per accedere ai servizi relativi a IMS (IMS DB, dialogo utente tramite MFS, ecc.).

## Registro dei programmi

Per partecipare e sfruttare tali costrutti, framework e servizi, i programmi Java modernizzati rispetto a quelli precedenti aderiscono a una struttura specifica documentata in [AWS Struttura Blu Age di un'applicazione modernizzata](#). All'avvio, AWS Blu Age Runtime raccoglierà tutti questi programmi in un «Registro dei programmi» comune in modo che possano essere richiamati (e richiamati) in seguito. Il Program Registry offre un accoppiamento libero e possibilità di scomposizione (poiché i programmi che si chiamano tra loro non devono essere modernizzati contemporaneamente).

## Ambienti di esecuzione

Sono disponibili ambienti e coreografie preesistenti:

- I batch basati su JCL, una volta modernizzati ai programmi Java e agli script Groovy, possono essere avviati in modo sincrono (bloccante) o asincrono (separato). In quest'ultimo caso, la loro esecuzione può essere monitorata tramite endpoint REST.
- Un sottosistema AWS Blu Age fornisce un ambiente di esecuzione simile a CICS attraverso:
  - un punto di ingresso utilizzato per avviare una transazione CICS ed eseguire i programmi associati nel rispetto della coreografia CICS «run levels»,
  - una memoria esterna per le definizioni delle risorse,

- un insieme omogeneo di API fluenti in Java che riproducono istruzioni, EXEC CICS
- un insieme di classi collegabili che riproducono servizi CICS, come Temporary Storage Queues, Temporary Data Queues o accesso ai file (di solito sono disponibili più implementazioni, come Amazon Managed Service per Apache Flink, Amazon Simple Queue Service o RabbitMQ per TD Queues),
- per le applicazioni rivolte agli utenti, il formato di descrizione dello schermo BMS è stato modernizzato in un'applicazione web Angular ed è supportata la corrispondente finestra di dialogo «pseudo-conversazionale».
- Allo stesso modo, un altro sottosistema fornisce coreografie basate su messaggi IMS e supporta la modernizzazione delle schermate dell'interfaccia utente in formato MFS.
- Inoltre, un terzo sottosistema consente l'esecuzione di programmi in un ambiente simile a iSeries, inclusa la modernizzazione delle schermate specificate da DSPF (Display File).

Tutti questi ambienti si basano su servizi comuni a livello di sistema operativo come:

- l'emulazione dell'allocazione e del layout della memoria legacy (Data Simplifier),
- riproduzione basata su thread Java dell'esecuzione delle «unità di esecuzione» COBOL e del meccanismo di passaggio dei parametri (istruzione), CALL
- emulazione di organizzazioni VSAM piatte e concatenate (attraverso il set di librerie Bluesam) e GDG Data Set,
- accesso agli archivi di dati, come RDBMS (dichiarazioni). EXEC SQL

## Apolidia e gestione delle sessioni

Una caratteristica importante di AWS Blu Age Runtime consiste nell'abilitare scenari di alta disponibilità (HA) e scalabilità orizzontale durante l'esecuzione di programmi modernizzati.

La pietra angolare di tutto ciò è l'assenza di stato, un esempio importante del quale è la gestione delle sessioni HTTP.

### Gestione delle sessioni

Poiché Tomcat è basato sul Web, un meccanismo importante a tal fine è la gestione delle sessioni HTTP (fornita da tomcat e Spring) e la progettazione senza stato. Pertanto, il design dell'apolidia si basa su quanto segue:

- gli utenti si connettono tramite HTTPS,
- i server delle applicazioni sono distribuiti con un sistema di bilanciamento del carico,
- quando un utente si connette per la prima volta all'applicazione, verrà autenticato e il server dell'applicazione creerà un identificatore (in genere all'interno di un cookie)
- questo identificatore verrà utilizzato come chiave per salvare e recuperare il contesto dell'utente da/verso una cache esterna (archivio dati).

La gestione dei cookie viene eseguita automaticamente dal framework AWS Blu Age e dal server tomcat sottostante, questo è trasparente per l'utente. Il browser Internet dell'utente lo gestirà automaticamente.

L'applicazione web Gapwalk può memorizzare lo stato della sessione (il contesto) in vari archivi di dati:

- Amazon ElastiCache per Redis
- Cluster Redis
- nella mappa di memoria (solo per ambienti di sviluppo e standalone, non adatto per HA).

## Alta disponibilità e apolidia

Più in generale, un principio di progettazione del framework AWS Blu Age è l'assenza di stato: la maggior parte degli stati non transitori necessari per riprodurre il comportamento dei programmi legacy non sono archiviati all'interno dei server delle applicazioni, ma condivisi attraverso una «singola fonte di verità» esterna e comune.

Esempi di tali stati sono le code di archiviazione temporanee o le definizioni delle risorse di CICS, e gli archivi esterni tipici per questi sono server o database relazionali compatibili con Redis.

Questo design, combinato con il bilanciamento del carico e le sessioni condivise, fa sì che la maggior parte delle finestre di dialogo rivolte all'utente (OLTP, «Online Transactional Processing») siano distribuibili tra più «nodi» (qui, istanze tomcat).

In effetti, un utente può eseguire una transazione su qualsiasi server e non importa se la chiamata di transazione successiva viene eseguita su un server diverso. Quindi, quando viene generato un nuovo server (a causa del ridimensionamento automatico o per sostituire un server non integro), possiamo garantire che qualsiasi server raggiungibile e integro possa eseguire la transazione come previsto con i risultati corretti (valore restituito previsto, modifica prevista dei dati nel database, ecc.).

## AWS Struttura Blu Age di un'applicazione modernizzata

Questo documento fornisce dettagli sulla struttura delle applicazioni modernizzate (utilizzando gli strumenti di refactoring di AWS Mainframe Modernization), in modo che gli sviluppatori possano svolgere varie attività, come:

- navigazione fluida tra le applicazioni.
- sviluppo di programmi personalizzati che possono essere richiamati dalle applicazioni modernizzate.
- rifattorizzazione sicura di applicazioni modernizzate.

Partiamo dal presupposto che tu abbia già conoscenze di base su quanto segue:

- concetti di codifica comuni e preesistenti, come record, set di dati e relative modalità di accesso ai record (indicizzata, sequenziale), VSAM, unità di esecuzione, script jcl, concetti CICS e così via.
- [codifica](#) java utilizzando il framework Spring.
- In tutto il documento, lo utilizziamo `short class names` per la leggibilità. Per ulteriori informazioni, vedete [AWS Mappature dei nomi completamente qualificate di Blu Age](#) per recuperare i nomi completi corrispondenti per gli elementi di runtime AWS Blu Age e [Mappature dei nomi completamente qualificate di terze parti](#) per recuperare i nomi completi corrispondenti per gli elementi di terze parti.
- [Tutti gli artefatti e i campioni vengono prelevati dai risultati del processo di modernizzazione dell'applicazione COBOL/CICS di esempio. CardDemo](#)

### Argomenti

- [Organizzazione degli artefatti](#)
- [Esecuzione e chiamata di programmi](#)
- [Scrivi il tuo programma](#)
- [Mappature dei nomi complete](#)

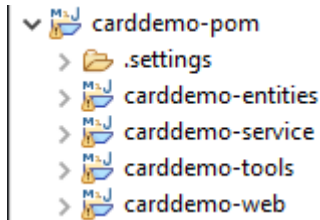
### Organizzazione degli artefatti

AWS Le applicazioni modernizzate di Blu Age sono confezionate come applicazioni web java (.war), che possono essere distribuite su un server JEE. [In genere, il server è un'istanza Tomcat che](#)



[incorpora il runtime AWS Blu Age Velocity, attualmente basato sui framework Springboot e Angular \(per la parte dell'interfaccia utente\).](#)

La guerra aggrega diversi artefatti componenti (.jar). Ogni jar è il risultato della compilazione (utilizzando lo strumento [maven](#)) di un progetto java dedicato i cui elementi sono il risultato del processo di modernizzazione.



L'organizzazione di base si basa sulla seguente struttura:

- Progetto Entities: contiene il modello di business e gli elementi di contesto. Il nome del progetto generalmente termina con «-entities». In genere, per un determinato programma COBOL legacy, ciò corrisponde alla modernizzazione della sezione I/O (set di dati) e alla divisione dei dati. È possibile avere più di un progetto di entità.
- Progetto di servizio: contiene elementi di modernizzazione della logica aziendale legacy. In genere, la divisione delle procedure di un programma COBOL. È possibile avere più di un progetto di servizio.
- Progetto di utilità: contiene strumenti e utilità comuni condivisi, utilizzati da altri progetti.
- Progetto Web: contiene la modernizzazione degli elementi relativi all'interfaccia utente, ove applicabile. Non utilizzato per progetti di modernizzazione esclusivamente in batch. Questi elementi dell'interfaccia utente potrebbero provenire dalle mappe CICS BMS, dai componenti IMS MFS e da altre fonti dell'interfaccia utente mainframe. È possibile avere più di un progetto Web.
































Entità: contenuti del progetto

#### Note

Le seguenti descrizioni si applicano solo agli output di modernizzazione COBOL e PL/I. Gli output di modernizzazione dei giochi di ruolo si basano su un layout diverso.

Prima di qualsiasi rifattorizzazione, l'organizzazione dei pacchetti nel progetto Entities è legata ai programmi modernizzati. È possibile farlo in un paio di modi diversi. Il modo preferito è utilizzare

il toolbox Refactoring, che funziona prima di attivare il meccanismo di generazione del codice. Si tratta di un'operazione avanzata, che viene spiegata nei corsi di formazione. BluAge Per ulteriori informazioni, vedere il workshop sul [refactoring](#). Questo approccio consente di preservare la capacità di rigenerare il codice java in un secondo momento, per beneficiare di ulteriori miglioramenti in futuro, ad esempio). L'altro modo è quello di eseguire regolari refactoring java, direttamente sul codice sorgente generato, utilizzando qualsiasi approccio di refactoring java che si desidera applicare, a proprio rischio e pericolo.

- ▼  src/main/java
  - ▼  aws.bluage.I3.workshop.cbact04c.business.context
    - >  Cbact04cConfiguration.java
    - >  Cbact04cContext.java
  - ▼  aws.bluage.I3.workshop.cbact04c.business.model
    - >  Abcode.java
    - >  AccountFile.java
    - >  AccountRecord.java
    - >  AcctfileStatus.java
    - >  ApplResult.java
    - >  CardXrefRecord.java
    - >  CobolTs.java
    - >  DiscgrpFile.java
    - >  DiscgrpStatus.java
    - >  DisGroupRecord.java
    - >  EndOfFile.java
    - >  ExternalParms.java
    - >  Group1.java
    - >  Group2.java
    - >  IoStatus.java
    - >  IoStatus04.java
    - >  TcatbalFile.java
    - >  TcatbalfStatus.java
    - >  Timing.java
    - >  TranCatBalRecord.java
    - >  TranfileStatus.java
    - >  TranRecord.java
    - >  TransactFile.java
    - >  WsCounters.java
    - >  WsMiscVars.java
    - >  XrefFile.java
    - >  XreffileStatus.java

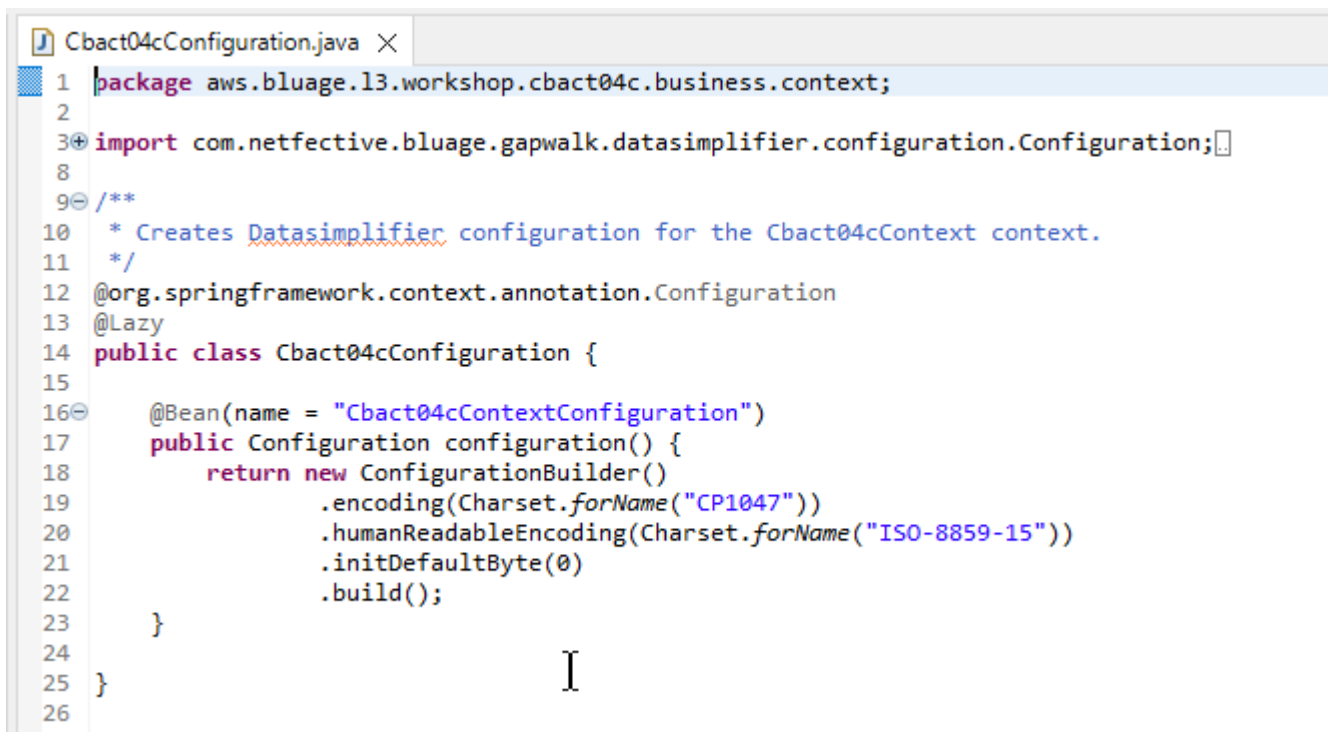
## Classi relative al programma

Ogni programma modernizzato è correlato a due pacchetti, un pacchetto business.context e un pacchetto business.model.

- *base package.program.business.context*

Il sottopacchetto `business.context` contiene due classi, una classe di configurazione e una classe `context`.

- Una classe di configurazione per il programma, che contiene dettagli di configurazione specifici per un determinato programma, come il set di caratteri da utilizzare per rappresentare gli elementi di dati basati su caratteri, il valore in byte predefinito per il riempimento degli elementi della struttura dei dati e così via. Il nome della classe termina con «Configurazione». È contrassegnato con l'annotation `@org.springframework.context.annotation.Configuration` e contiene un singolo metodo che deve restituire un `Configuration` oggetto configurato correttamente.



```
1 package aws.bluage.13.workshop.cbact04c.business.context;
2
3 import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
4
5
6
7
8
9 /**
10  * Creates Datasimplifier configuration for the Cbact04cContext context.
11  */
12 @org.springframework.context.annotation.Configuration
13 @Lazy
14 public class Cbact04cConfiguration {
15
16     @Bean(name = "Cbact04cContextConfiguration")
17     public Configuration configuration() {
18         return new ConfigurationBuilder()
19             .encoding(Charset.forName("CP1047"))
20             .humanReadableEncoding(Charset.forName("ISO-8859-15"))
21             .initDefaultByte(0)
22             .build();
23     }
24
25 }
26
```

- Una classe di contesto, che funge da ponte tra le classi di servizi del programma (vedi sotto) e le strutture di dati (`Record`) e i set di dati (`File`) del sottopacchetto del modello (vedi sotto). Il nome della classe termina con «Context» ed è una sottoclasse della `RuntimeContext` classe.

```

Cbact04cContext.java x
139 @Component("aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext")
140 @Import({
141     aws.bluage.l3.workshop.cbact04c.business.model.TcatbalFile.class
142     , aws.bluage.l3.workshop.cbact04c.business.model.XrefFile.class
143     , aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile.class
144     , aws.bluage.l3.workshop.cbact04c.business.model.AccountFile.class
145     , aws.bluage.l3.workshop.cbact04c.business.model.TransactFile.class
146 })
147 @Lazy
148 @Scope("prototype")
149 public class Cbact04cContext extends JicsRuntimeContext {
150
151     @Autowired
152     private TcatbalFile tcatbalFile;
153
154     @Autowired
155     private XrefFile xrefFile;
156
157     @Autowired
158     private DiscgrpFile discgrpFile;
159
160     @Autowired
161     private AccountFile accountFile;
162
163     @Autowired
164     private TransactFile transactFile;
165
166     private IndexedFile tcatbalFileFile;
167
168     private IndexedFile xrefFileFile;
169
170     private IndexedFile discgrpFileFile;
171
172     private IndexedFile accountFileFile;
173
174     private SequentialFile transactFileFile;
175
176     private TranCatBalRecord tranCatBalRecord;
177     private TcatbalfStatus tcatbalfStatus;
178     private CardXrefRecord cardXrefRecord;

```

- *base package.program.business.model*

Il sottopacchetto del modello contiene tutte le strutture di dati che il programma specificato può utilizzare. Ad esempio, qualsiasi struttura dati COBOL di livello 01 corrisponde a una classe nel sottopacchetto del modello (le strutture dati di livello inferiore sono proprietà della loro struttura a 01 livelli). Per ulteriori informazioni su come modernizziamo 01 strutture di dati, consulta.

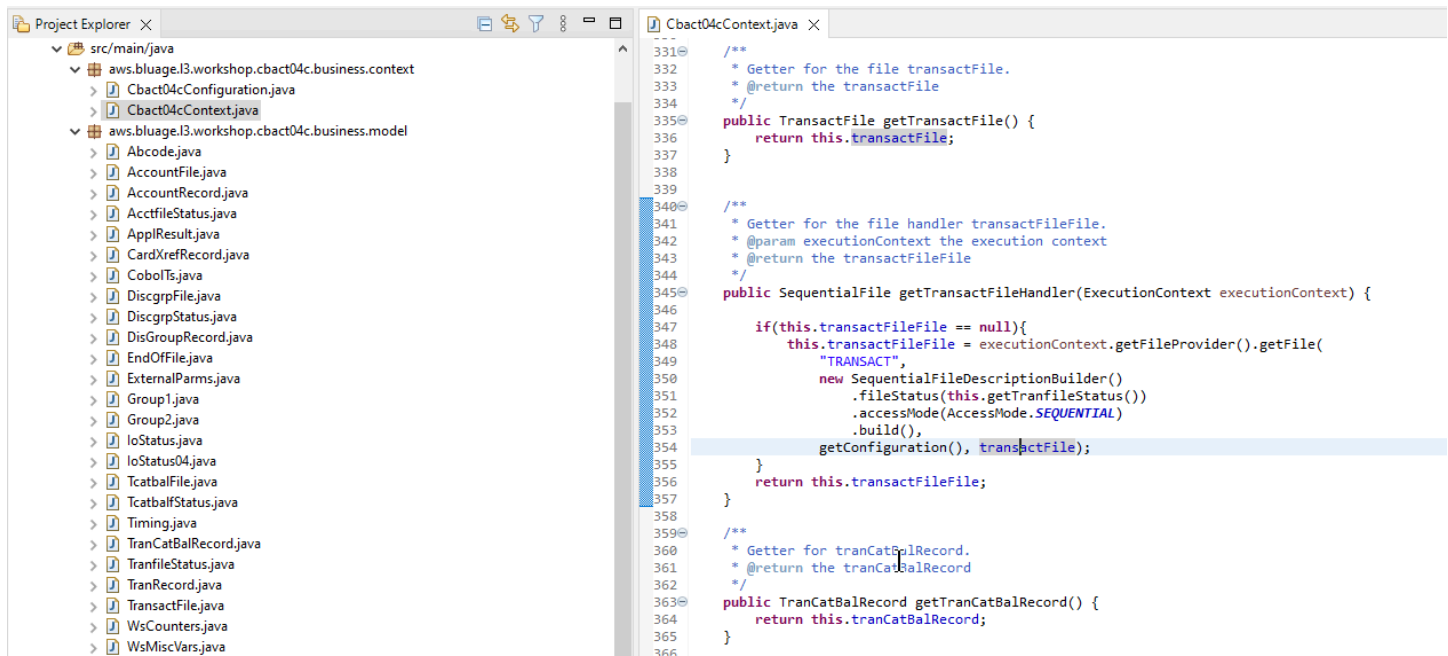
[Semplificatore di dati](#)

```

DiscgrpFile.java ×
1 package aws.bluage.l3.workshop.cbact04c.business.model;
2
3 import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
4 import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary;
5 import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group;
6 import com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference;
7 import com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference;
8 import com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity;
9 import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType;
10 import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType;
11 import org.springframework.beans.factory.annotation.Qualifier;
12 import org.springframework.context.annotation.Lazy;
13 import org.springframework.context.annotation.Scope;
14 import org.springframework.stereotype.Component;
15
16 /**
17  * Data simplifier file DiscgrpFile.
18  *
19  * <p>About 'fdDiscgrpRec' field, <br>uml entity: aws.bluage.l3.workshop.cbact04c.business.model.FdDiscgrpRec
20  * <br></p>
21  *
22  */
23 @Component("aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile")
24 @Lazy
25 @Scope("prototype")
26 public class DiscgrpFile extends RecordEntity {
27
28     private final Group root = new Group(getData());
29     private final Group fdDiscgrpRec = new Group(root);
30     private final Group fdDiscgrpKey = new Group(fdDiscgrpRec);
31     private final Elementary fdDisAcctGroupId = new Elementary(fdDiscgrpKey, new AlphanumericType(10));
32     private final Elementary fdDisTranTypeCd = new Elementary(fdDiscgrpKey, new AlphanumericType(2));
33     private final Elementary fdDisTranCatCd = new Elementary(fdDiscgrpKey, new ZonedType(4, 0, false));
34     private final Elementary fdDiscgrpData = new Elementary(fdDiscgrpRec, new AlphanumericType(34));
35

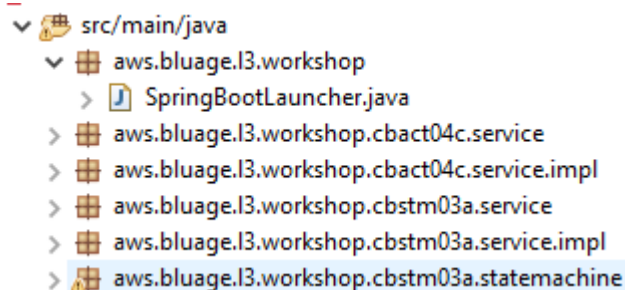
```

Tutte le classi estendono la RecordEntity classe, che rappresenta l'accesso a una rappresentazione di record aziendali. Alcuni record hanno uno scopo speciale, in quanto sono legati a unFile. L'associazione tra a Record e a File viene effettuata nei corrispondenti FileHandler metodi \* trovati nella classe context durante la creazione dell'oggetto file. Ad esempio, l'elenco seguente mostra come TransactfileFile File è associato a TransactFile Record (dal sottopacchetto del modello).



## Contenuto del progetto di servizio

Ogni progetto di servizio viene fornito con un'applicazione [Springboot](#) dedicata, che viene utilizzata come spina dorsale dell'architettura. Ciò si concretizza attraverso la classe denominata `SpringBootLauncher`, che si trova nel pacchetto base del servizio java sources:



Questa classe è in particolare responsabile di:

- creare un collante tra le classi del programma e le risorse gestite (fonti di dati/gestori di transazioni/mappature dei set di dati/ecc...).
- fornire un ai programmi. `ConfigurableApplicationContext`
- scoprendo tutte le classi contrassegnate come componenti a molla (`@Component`).
- assicurando che i programmi siano registrati correttamente nel `ProgramRegistry` -- vedi il metodo di inizializzazione responsabile di questa registrazione.

```
/**
 * Initialization method called when the spring application is ready.
 * Register all programs and services to the gapwalk shared context.
 * @param event the application ready event
 */
@EventListener
public void initialize(ApplicationReadyEvent event) {
    Map<String, ProgramContainer> programContainers = event.getApplicationContext().getBeansOfType(ProgramContainer.class);
    programContainers.values().forEach(ProgramRegistry::registerProgram);
    Map<String, ServiceContainer> serviceContainers = event.getApplicationContext().getBeansOfType(ServiceContainer.class);
    serviceContainers.values().forEach(ServiceRegistry::registerService);
}
```

## Artefatti relativi al programma

Senza un precedente refactoring, i risultati della modernizzazione della logica aziendale sono organizzati su due o tre pacchetti per programma precedente:

- aws.bluage.I3.workshop.cocrdslc.service
    - CocrdslcProcess.java
      - CocrdslcProcess
        - cocrdslc(CocrdslcContext, ExecutionController) : void
        - commonReturn(CocrdslcContext, ExecutionController) : void
        - editAccount(CocrdslcContext, ExecutionController) : void
        - editCard(CocrdslcContext, ExecutionController) : void
        - editMapInpputs(CocrdslcContext, ExecutionController) : void
        - getcardByacct(CocrdslcContext, ExecutionController) : void
        - getcardByacctcard(CocrdslcContext, ExecutionController) : void
        - processInpputs(CocrdslcContext, ExecutionController) : void
        - receiveMap(CocrdslcContext, ExecutionController) : void
        - screenInit(CocrdslcContext, ExecutionController) : void
        - sendLongText(CocrdslcContext, ExecutionController) : void
        - sendMap(CocrdslcContext, ExecutionController) : void
        - sendPlainText(CocrdslcContext, ExecutionController) : void
        - sendScreen(CocrdslcContext, ExecutionController) : void
        - setupScreenAttrs(CocrdslcContext, ExecutionController) : void
        - setupScreenVars(CocrdslcContext, ExecutionController) : void
        - yyyyStorePffkey(CocrdslcContext, ExecutionController) : void
  - aws.bluage.I3.workshop.cocrdslc.service.impl
    - CocrdslcProcessImpl.java
      - CocrdslcProcessImpl
        - LOGGER
        - cocrdslcProcedureDivisionStateMachineRunner
        - cocrdslc(CocrdslcContext, ExecutionController) : void
        - commonReturn(CocrdslcContext, ExecutionController) : void
        - editAccount(CocrdslcContext, ExecutionController) : void
        - editCard(CocrdslcContext, ExecutionController) : void
        - editMapInpputs(CocrdslcContext, ExecutionController) : void
        - getcardByacct(CocrdslcContext, ExecutionController) : void
        - getcardByacctcard(CocrdslcContext, ExecutionController) : void
        - processInpputs(CocrdslcContext, ExecutionController) : void
        - receiveMap(CocrdslcContext, ExecutionController) : void
        - screenInit(CocrdslcContext, ExecutionController) : void
        - sendLongText(CocrdslcContext, ExecutionController) : void
        - sendMap(CocrdslcContext, ExecutionController) : void
        - sendPlainText(CocrdslcContext, ExecutionController) : void
        - sendScreen(CocrdslcContext, ExecutionController) : void
        - setupScreenAttrs(CocrdslcContext, ExecutionController) : void
        - setupScreenVars(CocrdslcContext, ExecutionController) : void
        - yyyyStorePffkey(CocrdslcContext, ExecutionController) : void
  - aws.bluage.I3.workshop.cocrdslc.statemachine
    - CocrdslcProcedureDivisionStateMachineController.java
      - CocrdslcProcedureDivisionStateMachineController
        - Events
        - States
          - stateProcess
          - configureStateMachine(StateMachineStateConfigurer<States, Events>, StateMachineTransitionConfigurer<States, Events>) : void
          - configureStateMachine(StateMachineStateConfigurer<States, Events>, StateMachineTransitionConfigurer<States, Events>, RuntimeContext, ExecutionController) : void
          - configureTransitions(StateMachineTransitionConfigurer<States, Events>) : void
    - CocrdslcProcedureDivisionStateMachineService.java
      - CocrdslcProcedureDivisionStateMachineService
        - LOGGER
        - bluesamManager
        - instanceCocrdslcProcess
        - instanceStateMachineController
        - \_0000Main(CocrdslcContext, ExecutionController) : void
        - abendRoutine(CocrdslcContext, ExecutionController) : void

Il caso più esaustivo avrà tre pacchetti:



- *base package.program.service*: contiene un'interfaccia denominata Program Process, che utilizza metodi aziendali per gestire la logica aziendale, preservando il precedente flusso di controllo dell'esecuzione.
- *base package.program.service.impl*: contiene una classe denominata ProgramProcessImpl, che è l'implementazione dell'interfaccia Process descritta in precedenza. È qui che le istruzioni precedenti vengono «tradotte» in istruzioni java, basandosi sul framework AWS Blu Age:

```

CocrdslcProcessImpl.java X
210  /**
211   * Process operation sendScreen.
212   *
213   * @param ctx
214   * @param ctrl
215   */
216  @Override
217  public void sendScreen(final CocrdslcContext ctx, final ExecutionController ctrl) {
218      ctx.getCcWorkAreas().setCcardNextMapset(ctx.getWsLiterals().getLitThismapset());
219      ctx.getCcWorkAreas().setCcardNextMap(ctx.getWsLiterals().getLitThismap());
220      ctx.getCarddemoCommarea().setCdemoPgmReenter(true);
221      SendMapBuilder.newInstance(ctx.getDfheiblk(), ctx)
222          .withMap(ctx.getCcWorkAreas().getCcardNextMap())
223          .withMapset(ctx.getCcWorkAreas().getCcardNextMapset())
224          .withData(ctx.getGroup1().getCcrdslaoReference())
225          .withCursor()
226          .withErase()
227          .withFreeKB()
228          .execute();
229      ctx.getWsMiscStorage().setWsRespCd(ctx.getDfheiblk().getEibresp());
230  }
231
232  /**
233   * Process operation processInputs.
234   *
235   * @param ctx
236   * @param ctrl
237   */
238  @Override
239  public void processInputs(final CocrdslcContext ctx, final ExecutionController ctrl) {
240      receiveMap(ctx, ctrl);
241      editMapInputs(ctx, ctrl);
242      ctx.getCcWorkAreas().setCcardErrorMsg(ctx.getWsMiscStorage().getWsReturnMsg());
243      ctx.getCcWorkAreas().setCcardNextProg(ctx.getWsLiterals().getLitThispgm());
244      ctx.getCcWorkAreas().setCcardNextMapset(ctx.getWsLiterals().getLitThismapset());
245      ctx.getCcWorkAreas().setCcardNextMap(ctx.getWsLiterals().getLitThismap());
246  }
247

```

- *base package.program.statemachine*: questo pacchetto potrebbe non essere sempre presente. È necessario quando la modernizzazione del flusso di controllo legacy deve utilizzare un approccio basato su una macchina a stati (in particolare utilizzando il [State Machine framework Spring](#)) per coprire adeguatamente il flusso di esecuzione precedente.

In tal caso, il sottopacchetto `statemachine` contiene due classi:

- **ProgramProcedureDivisionStateMachineController**: una classe che estende una classe che implementa le interfacce `StateMachineController` (definisce le operazioni necessarie per controllare l'esecuzione di una macchina a stati) e `StateMachineRunner` (definisce le operazioni necessarie per eseguire una macchina a stati), utilizzate per guidare la meccanica della macchina a stati Spring; ad esempio, come nel `SimpleStateMachineController` caso di esempio.

```

1 package aws.bluage.13.workshop.cocrdslc.statemachine;
2
3 import aws.bluage.13.workshop.cocrdslc.business.context.CocrdslcContext;
4
5 /**
6  * Controller managing the state machine "CocrdslcProcedureDivisionStateMachine" execution.
7  */
8 @Component("aws.bluage.13.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineController")
9 @Import({
10     aws.bluage.13.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineService.class
11 })
12 @Lazy
13 public class CocrdslcProcedureDivisionStateMachineController extends SimpleStateMachineController<States, Events> {
14
15     /**
16      * State machine states.
17      */
18     public enum States {
19         _0000_MAIN_1, _0000_MAIN, ABEND_ROUTINE, FINAL, LOCAL_FINAL
20     }
21
22     /**
23      * State machine events.
24      */
25     public enum Events {
26         TO_0000_MAIN_1, TO_0000_MAIN, TO_ABEND_ROUTINE, TO_FINAL, TO_LOCAL_FINAL
27     }
28
29     /**
30      * State machine state process service provider.
31      */
32     @Autowired
33     @Lazy
34     private CocrdslcProcedureDivisionStateMachineService stateProcess;
35
36     @Override
37     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
38         throw new UnsupportedOperationException("Please use the four arguments configureStateMachine method instead: configureStateMachine(StateMachineStateConfigurer<States, Events> states, "
39             + "StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl)");
40     }
41
42     @Override
43     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl) throws Exception {
44         StateConfigurer<States, Events> configurer = states.withStates();
45         configurer.initial(States._0000_MAIN_1).end(States.FINAL);
46         configurer.state(States._0000_MAIN_1);
47         configurer.state(States.FINAL);
48
49         StateConfigurer<States, Events> subConfigurer = states.withStates().parent(States._0000_MAIN_1);
50         subConfigurer.initial(States._0000_MAIN).end(States.LOCAL_FINAL);
51         CocrdslcContext lctx = (CocrdslcContext) ctx;
52         subConfigurer.state(States._0000_MAIN, buildAction(() -> {stateProcess._0000Main(lctx, ctrl);}), null);
53         subConfigurer.state(States.ABEND_ROUTINE, buildAction(() -> {stateProcess.abendRoutine(lctx, ctrl);}), null);
54
55         configureTransitions(transitions);
56     }
57
58     /**
59      * Declare state machine transitions.
60      * @param transitions the transitions configuration helper
61      */
62     private void configureTransitions(StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
63         transitions.withLocal().source(States._0000_MAIN_1).target(States.ABEND_ROUTINE).event(Events.TO_ABEND_ROUTINE);
64         transitions.withExternal().source(States.ABEND_ROUTINE).target(States.FINAL).event(Events.TO_FINAL);
65     }
66 }
67
68
69
70
71
72
73
74
75
76
77
78
79
80

```

Il controller della macchina a stati definisce i possibili diversi stati e le transizioni tra di essi, il che riproduce il flusso di controllo dell'esecuzione precedente per un determinato programma.

Durante la creazione della macchina a stati, il controller fa riferimento ai metodi definiti nella classe di servizio associata situata nel pacchetto della macchina a stati e descritti di seguito:

```

subConfigurer.state(States._0000_MAIN, buildAction(() ->
    {stateProcess._0000Main(lctx, ctrl);}), null);

```

```
subConfigurer.state(States.ABEND_ROUTINE, buildAction(() ->
    {stateProcess.abendRoutine(lctx, ctrl);}), null);
```

- *ProgramProcedureDivisionStateMachineService*: questa classe di servizio rappresenta una logica aziendale che deve essere associata alla macchina a stati creata dal controller della macchina a stati, come descritto in precedenza.

Il codice nei metodi di questa classe utilizza gli eventi definiti nel controller della macchina a stati:

```
CocrdslcProcedureDivisionStateMachineService.java ×
59  /**
60   * State process operation _0000Main.
61   *
62   * @param ctx
63   * @param ctrl
64   */
65  void _0000Main(CocrdslcContext ctx, ExecutionController ctrl) {
66      ctx.getDfheiblk().bind(ArgUtils.get(ctx, 0));
67      ctx.getDfhcommarea().bind(ArgUtils.get(ctx, 1));
68
69      /*
70      *****
71      Program:      COCRDSL.CBL
72      Layer:       Business logic
73      Function:    Accept and process credit card detail request
74      *****
75      Copyright Amazon.com, Inc. or its affiliates.
76      All Rights Reserved.
77      Licensed under the Apache License, Version 2.0 (the "License").
78      You may not use this file except in compliance with the License.
79      You may obtain a copy of the License at
80      http://www.apache.org/licenses/LICENSE-2.0
81      Unless required by applicable law or agreed to in writing,
82      software distributed under the License is distributed on an
83      "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
84      either express or implied. See the License for the specific
85      language governing permissions and limitations under the License
86      *****
87      Ver: CardDemo v1.0-15-g27d6c6f-68 Date: 2022-07-19 23:16:00 CDT */
88      instanceStateMachineController.registerSignalHandler(Events.TO_ABEND_ROUTINE, "!ABEND");
89      HandleAbendBuilder.newInstance(ctx.getDfheiblk(), ctx).execute().handleException();
90      ctx.getCcWorkAreas().getCcWorkAreaReference().getField().initialize();
91      ctx.getWsMiscStorage().getField().initialize();
92      DataUtils.initialize(ctx.getWsCommarea().getWsCommareaReference());
93  }
```

```

CocrdslcProcedureDivisionStateMachineService.java X
221      *
222      * @param ctx
223      * @param ctrl
224      */
225  void abendRoutine(CocrdslcContext ctx, ExecutionController ctrl) {
226      if (DataUtils.isLowValue(ctx.getAbendData().getAbendMsgReference())) {
227          ctx.getAbendData().setAbendMsg("UNEXPECTED ABEND OCCURRED.");
228      }
229      ctx.getAbendData().setAbendCulprit(ctx.getWsLiterals().getLitThispgm());
230      SendTextBuilder.newInstance(ctx.getDfheiblk(), ctx)
231          .withData(ctx.getAbendData())
232          .withLength(134)
233          .execute();
234      HandleAbendBuilder.newInstance(ctx.getDfheiblk(), ctx).cancel().execute().handleException();
235      AbendBuilder.newInstance(ctx.getDfheiblk(), ctx).withAbendCode("9999").execute().handleException();
236
237      /*
238      Ver: CardDemo v1.0-15-g27d6c6f-68 Date: 2022-07-19 23:12:33 CDT */
239      instanceStateMachineController.sendEvent(Events.TO_FINAL);
240
241  }
242  }
243

```

Il servizio statemachine effettua anche chiamate all'implementazione del servizio di processo descritta in precedenza:

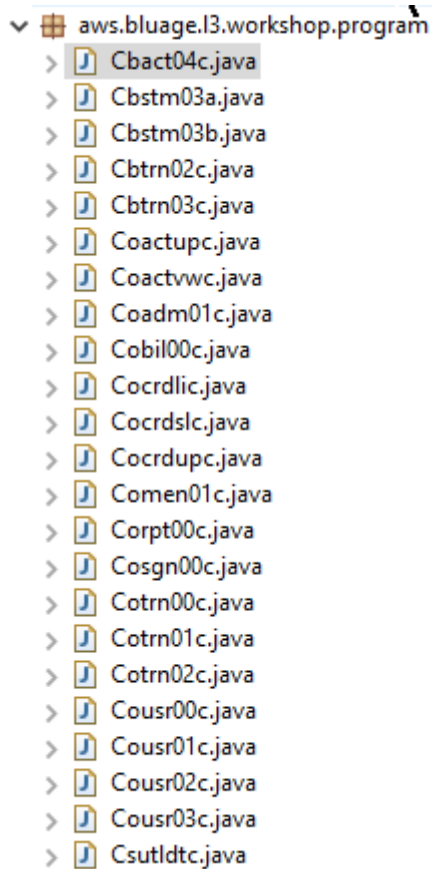
```

CocrdslcProcedureDivisionStateMachineService.java X
166      /*
167      .....
168      COMING FROM CREDIT CARD LIST SCREEN
169      SELECTION CRITERIA ALREADY VALIDATED
170      ..... */
171  } else if (ctx.getCarddemoCommarea().isCdemoPgmEnter() && DataUtils.compare(ctx.getCarddemoCommarea().getCdemoFromProgramReference(), ctx.getWsLiterals().getLitCclistpgmReference()) == 0) {
172      ctx.getWsMiscStorage().setInputOk(true);
173      ctx.getCWorkAreas().setCcAcctIdN(ctx.getCarddemoCommarea().getCdemoAcctId());
174      ctx.getCWorkAreas().setCcCardNumN(ctx.getCarddemoCommarea().getCdemoCardNum());
175      instanceCocrdslcProcess.getCardByacctcard(ctx, ctrl);
176      instanceCocrdslcProcess.sendMap(ctx, ctrl);
177      instanceCocrdslcProcess.commonReturn(ctx, ctrl);
178  } else if (ctx.getCarddemoCommarea().isCdemoPgmEnter()) {
179
180      /*
181      .....
182      COMING FROM SOME OTHER CONTEXT
183      SELECTION CRITERIA TO BE GATHERED
184      ..... */
185      instanceCocrdslcProcess.sendMap(ctx, ctrl);
186      instanceCocrdslcProcess.commonReturn(ctx, ctrl);
187  } else if (ctx.getCarddemoCommarea().isCdemoPgmReenter()) {
188      instanceCocrdslcProcess.processInputs(ctx, ctrl);
189      if (ctx.getWsMiscStorage().isInputError()) {
190          instanceCocrdslcProcess.sendMap(ctx, ctrl);
191          instanceCocrdslcProcess.commonReturn(ctx, ctrl);
192      } else {
193          instanceCocrdslcProcess.getCardByacctcard(ctx, ctrl);
194          instanceCocrdslcProcess.sendMap(ctx, ctrl);
195          instanceCocrdslcProcess.commonReturn(ctx, ctrl);
196      }
197  }
198  } else {
199      ctx.getAbendData().setAbendCulprit(ctx.getWsLiterals().getLitThispgm());
200      ctx.getAbendData().setAbendCode("0001");
201      DataUtils.setToBlank(ctx.getAbendData().getAbendReasonReference());
202      ctx.getWsMiscStorage().setWsReturnMsg("UNEXPECTED DATA SCENARIO");
203      instanceCocrdslcProcess.sendPlainText(ctx, ctrl);
204  }
205
206      /*
207      If we had an error setup error message that slipped through
208      Display and return */
209  if (ctx.getWsMiscStorage().isInputError()) {
210      ctx.getCWorkAreas().setCardErrorMsg(ctx.getWsMiscStorage().getWsReturnMsg());
211      instanceCocrdslcProcess.sendMap(ctx, ctrl);
212      instanceCocrdslcProcess.commonReturn(ctx, ctrl);
213  }
214  instanceCocrdslcProcess.commonReturn(ctx, ctrl);
215

```

Inoltre, un pacchetto denominato *base package*.program svolge un ruolo importante, in quanto raccoglie una classe per programma, che fungerà da punto di ingresso al programma (maggiori

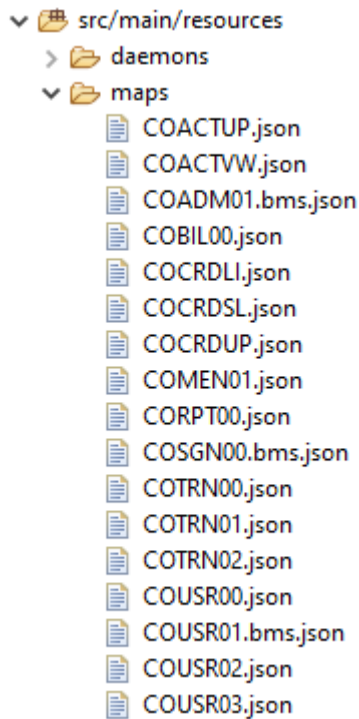
dettagli su questo argomento più avanti). Ogni classe implementa l'Programinterfaccia, indicatore di un punto di ingresso del programma.



## Altri manufatti

- Compagni BMS MAP

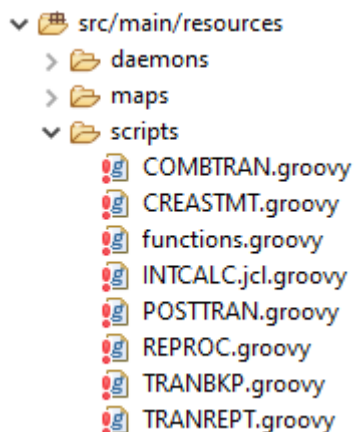
Oltre agli elementi relativi al programma, il progetto di servizio può contenere altri artefatti per vari scopi. Nel caso della modernizzazione di un'applicazione online CICS, il processo di modernizzazione produce un file json e lo inserisce nella cartella map della cartella /src/main/resources:



Il runtime Blu Age utilizza quei file json per associare i record utilizzati dall'istruzione SEND MAP ai campi dello schermo.

- Groovy Scripts

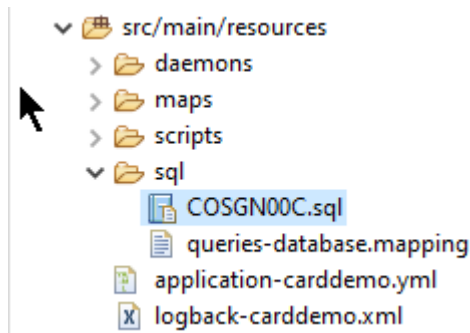
Se l'applicazione precedente aveva script JCL, questi sono stati modernizzati come script [groovy](#), [archiviati nella cartella /src/main/resources/scripts](#) (più su quella posizione specifica più avanti):



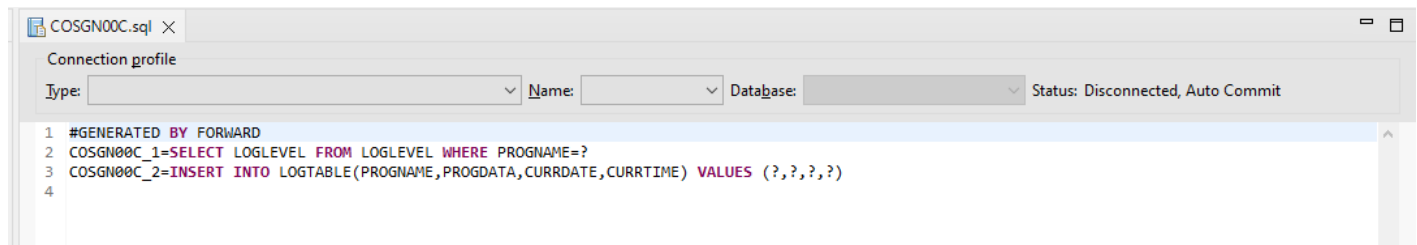
Questi script vengono utilizzati per avviare processi in batch (carichi di lavoro di elaborazione dati dedicati, non interattivi e che richiedono un uso intensivo della CPU).

- File SQL

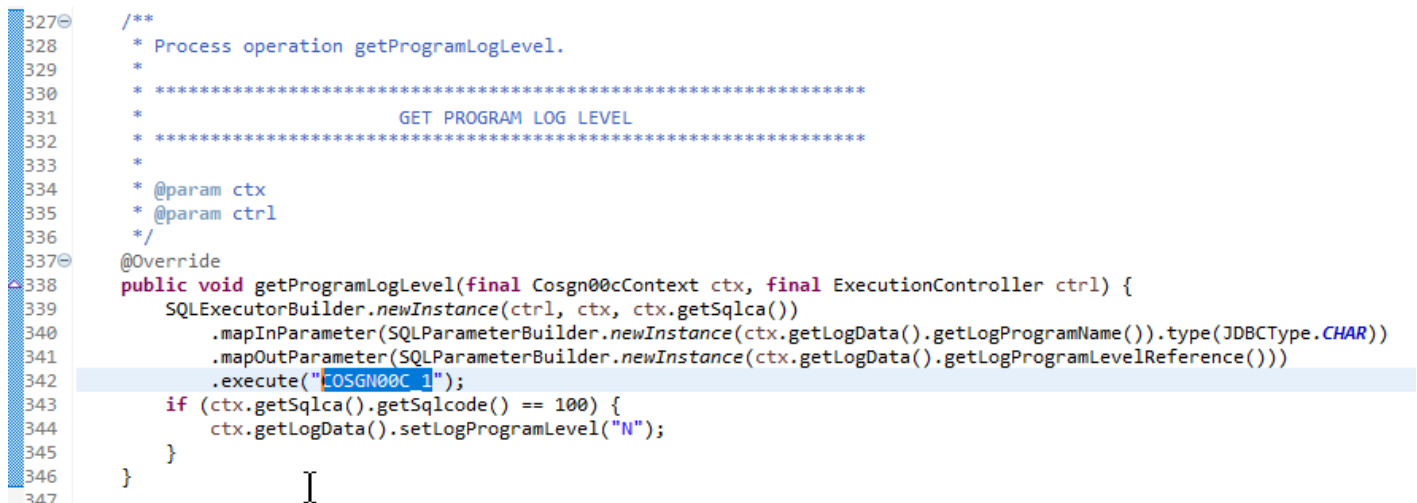
Se l'applicazione precedente utilizzava query SQL, le corrispondenti query SQL modernizzate sono state raccolte in file di proprietà dedicati, con lo schema di denominazione program .sql, dove program è il nome del programma che utilizza tali query.



Il contenuto di questi file sql è una raccolta di voci (key=query), in cui ogni query è associata a una chiave univoca, che il programma modernizzato utilizza per eseguire la query specificata:



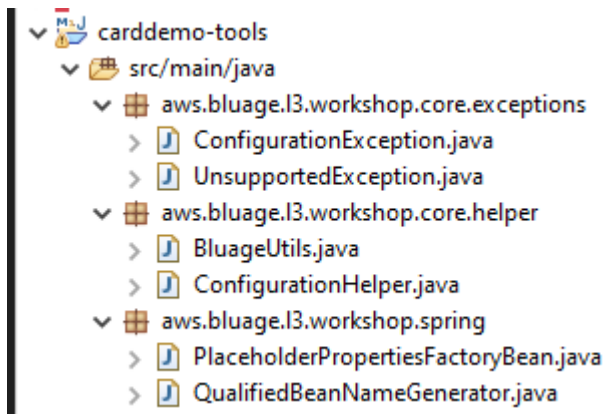
Ad esempio, il programma COSGN00C sta eseguendo la query con la chiave «COSGN00C\_1» (la prima voce nel file sql):





## Utilità, contenuti del progetto

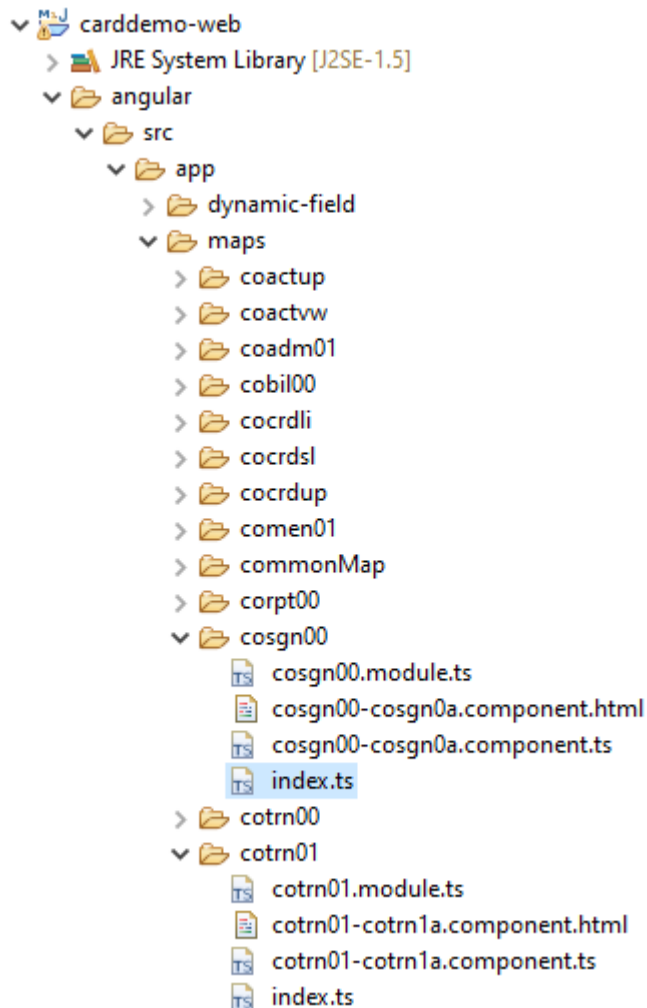
Il progetto utilities, il cui nome termina con «-tools», contiene una serie di utilità tecniche che potrebbero essere utilizzate da tutti gli altri progetti.



## Contenuti dei progetti Web

Il progetto web è presente solo durante la modernizzazione degli elementi dell'interfaccia utente legacy. [I moderni elementi dell'interfaccia utente utilizzati per creare il front-end dell'applicazione modernizzato sono basati su Angular.](#) L'applicazione di esempio utilizzata per mostrare gli artefatti della modernizzazione è un'applicazione COBOL/CICS, in esecuzione su un mainframe. Il sistema CICS utilizza MAP per rappresentare le schermate dell'interfaccia utente. Gli elementi moderni corrispondenti saranno, per ogni mappa, un file html accompagnato da file [Typescript](#):





Il progetto web si occupa solo dell'aspetto frontend dell'applicazione. Il progetto di servizio, che si basa sui progetti di utilità ed entità, fornisce i servizi di backend. Il collegamento tra il frontend e il backend viene effettuato tramite l'applicazione web denominata Gapwalk-Application, che fa parte della distribuzione di runtime Blu Age standard. AWS

## Esecuzione e chiamata di programmi

Nei sistemi precedenti, i programmi vengono compilati come eseguibili autonomi che possono richiamare se stessi tramite un meccanismo CALL, ad esempio l'istruzione COBOL CALL, passando argomenti quando necessario. Le applicazioni modernizzate offrono le stesse funzionalità ma utilizzano un approccio diverso, poiché la natura degli artefatti coinvolti è diversa da quelle precedenti.

Sul versante modernizzato, i punti di ingresso del programma sono classi specifiche che implementano l'Programinterfaccia, sono componenti Spring (@Component) e si trovano nei progetti di servizio, in un pacchetto denominato *base package.program*

## Registrazione dei programmi

Ogni volta che viene avviato il server [Tomcat](#) che ospita le applicazioni modernizzate, viene avviata anche l'applicazione Springboot di servizio, che attiva la registrazione dei programmi. Un registro dedicato denominato `ProgramRegistry` è popolato da voci di programma, ciascuna delle quali viene registrata utilizzando i relativi identificatori, una voce per identificatore di programma noto, il che significa che se un programma è conosciuto con diversi identificatori, il registro contiene tante voci quanti sono gli identificatori.

La registrazione per un determinato programma si basa sulla raccolta di identificatori restituiti dal metodo `()`: `getProgramIdentifiers`

```

Cbact04c.java x
1  package aws.bluage.l3.workshop.program;
2
3  import aws.bluage.l3.workshop.SpringBootLauncher;
24
25 /**
26  * Reference the spring application of program CBACT04C.
27  * Provides an access to the contained program for the run unit.
28  */
29  @Component
30  @Import({
31  aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cConfiguration.class,
32  aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext.class,
33  aws.bluage.l3.workshop.cbact04c.service.impl.Cbact04cProcessImpl.class
34  })
35  public class Cbact04c implements Program {
36  /**
37  * Unique identifiers for the contained program.
38  */
39  private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("CBACT04C").collect(Collectors.toSet()));
40
41  /**
42  * Main program identifier for the contained program.
43  */
44  private static final String programIdentifier = "CBACT04C";
45  @Autowired
46  PlatformTransactionManager transactionManager;
47
48  @Autowired
49  Map<String, DataSource> datasources;
50  @Autowired
51  BeanFactory beanFactory;
52  /**
53  * {@inheritDoc}
54  */
55  @Override
56  public ConfigurableApplicationContext getSpringApplication() {
57  return SpringBootLauncher.getCac();
58  }
59
60  /**
61  * {@inheritDoc}
62  */
63  @Override
64  public void updateExecutionContext(ExecutionContext executionContext) {
65  executionContext.setDatasources(datasources);
66  executionContext.setDatabaseSupport(ExecutionContext.DatabaseSupport.POSTGRE);
67  executionContext.setSqlcaVersion(ExecutionContext.SqlcaVersion.getEnum("ansi-comp5"));
68  executionContext.setTransactionManager(transactionManager);
69  executionContext.setUseSQLDateNewParadigm(true);
70  executionContext.setUseSQLTrimStringType(false);
71  }
72
73  /**
74  * {@inheritDoc}
75  */
76  @Override
77  public Set<String> getProgramIdentifiers() {
78  return programIdentifiers;
79  }
80

```

In questo esempio, il programma viene registrato una sola volta, con il nome 'CBACT04C' (guardate il contenuto della collezione ProgramIdentifiers). I log di tomcat mostrano ogni registrazione del programma. La registrazione del programma dipende solo dagli identificatori di programma dichiarati e non dal nome della classe del programma stesso (sebbene in genere gli identificatori di programma e i nomi delle classi di programma siano allineati).

Lo stesso meccanismo di registrazione si applica ai programmi di utilità forniti dalle varie applicazioni web di utilità AWS Blu Age, che fanno parte della distribuzione di runtime AWS Blu Age. Ad esempio, la webapp Gapwalk-Utility-Pgm fornisce gli equivalenti funzionali delle utilità di sistema

z/OS (IDCAMS, ICEGENER, SORT e così via) e può essere richiamata da programmi o script modernizzati. Tutti i programmi di utilità disponibili registrati all'avvio di Tomcat vengono registrati nei registri di Tomcat.

## Registrazione di script e demoni

Un processo di registrazione simile, all'avvio di Tomcat, si verifica per gli script groovy che si trovano nella gerarchia delle cartelle `/src/main/resources/scripts`. La gerarchia delle cartelle degli script viene attraversata e tutti gli script groovy scoperti (tranne lo script riservato `special.functions.groovy`) vengono registrati in `ScriptRegistry`, utilizzando il loro nome breve (la parte del nome del file di script che si trova prima del primo carattere punto) come chiave per il recupero.

### Note

- Se diversi script hanno nomi di file che generano la stessa chiave di registrazione, viene registrata solo la più recente, sovrascrivendo qualsiasi registrazione rilevata in precedenza per quella determinata chiave.
- Considerando la nota precedente, prestate attenzione quando utilizzate le sottocartelle poiché il meccanismo di registrazione appiattisce la gerarchia e potrebbe portare a sovrascritture impreviste. La gerarchia non conta nel processo di registrazione: in genere `/scripts/a/myscript.groovy` e `/scripts/b/myscript.groovy` porteranno a `/scripts/b/myscript.groovy` sovrascrivere `/scripts/a/myscript.groovy`.

Gli script groovy nella cartella `/src/main/resources/daemons` vengono gestiti in modo leggermente diverso. Sono ancora registrati come normali script, ma in aggiunta vengono lanciati una sola volta, direttamente all'avvio di Tomcat, in modo asincrono.

Dopo la registrazione degli script in `ScriptRegistry`, una chiamata REST può avviarli, utilizzando gli endpoint dedicati esposti dall'applicazione Gapwalk. Per ulteriori informazioni, consultate la documentazione corrispondente.

## Programmi che chiamano programmi

Ogni programma può richiamare un altro programma come sottoprogramma, passandogli dei parametri. A tale scopo, i programmi utilizzano un'implementazione dell'`ExecutionController`interfaccia (nella maggior parte dei casi si tratta di un'`ExecutionControllerImpl`istanza), oltre a un meccanismo API fluido chiamato `the CallBuilder` per creare gli argomenti delle chiamate al programma.

Tutti i metodi dei programmi accettano `RuntimeContext` sia a che an `ExecutionController` come argomenti del metodo, quindi an `ExecutionController` è sempre disponibile per chiamare altri programmi.

Si veda, ad esempio, il diagramma seguente, che mostra come il programma CBST03A chiama il programma CBST03B come sottoprogramma, passandogli dei parametri:

```

Cbstm03aProcessImpl.java x
67  /**
68   * Process operation xreffileGetNext.
69   *
70   * -----*
71   *
72   * @param ctx
73   * @param ctrl
74   */
75  @Override
76  public void xreffileGetNext(final Cbstm03aContext ctx, final ExecutionController ctrl) {
77      ctx.getWsM03bArea().setWsM03bDd("XREFFILE");
78      ctx.getWsM03bArea().setM03bRead(true);
79      DataUtils.setToZeroes(ctx.getWsM03bArea().getWsM03bRcReference());
80      DataUtils.setToBlank(ctx.getWsM03bArea().getWsM03bFldtReference());
81      ctrl.callSubProgram("CBSTM03B", CallBuilder.newInstance(
82          .byReference(ctx.getWsM03bArea())
83          .getArguments(), ctx);
84      if (DataUtils.compare(ctx.getWsM03bArea().getWsM03bRcReference(), "00") == 0) {
85
86          /*
87           Do nothing */
88      } else if (DataUtils.compare(ctx.getWsM03bArea().getWsM03bRcReference(), "10") == 0) {
89          ctx.getMiscVariables().setEndOfFile("Y");
90      } else {
91          if (LOGGER.isInfoEnabled()) LOGGER.info("ERROR READING XREFFILE");
92          if (LOGGER.isInfoEnabled()) LOGGER.info("{}{}", "RETURN CODE: ", ctx.getWsM03bArea().getWsM03bRc());
93         abendProgram(ctx, ctrl);
94      }
95      ctx.getCardXrefRecord().setBytes(ctx.getWsM03bArea().getWsM03bFldtReference().getBytes());
96  }
97

```

- Il primo argomento di `ExecutionController.callSubProgram` è un identificatore del programma da chiamare (ovvero uno degli identificatori utilizzati per la registrazione del programma, vedere i paragrafi precedenti).
- Il secondo argomento, che è il risultato della compilazione di `CallBuilder`, è una matrice di `Record`, corrispondente ai dati passati da chiamante a chiamato.
- Il terzo e ultimo argomento è l'istanza del chiamante `RuntimeContext`.

Tutti e tre gli argomenti sono obbligatori e non possono essere nulli, ma il secondo argomento può essere una matrice vuota.

Il chiamante sarà in grado di gestire i parametri passati solo se originariamente è stato progettato per farlo. Per un programma COBOL precedente, ciò significa avere una sezione LINKAGE e una clausola USING per la divisione delle procedure per utilizzare gli elementi LINKAGE.

[Ad esempio, consultate il file sorgente COBOL CBSTM03B.CBL corrispondente:](#)

[github.com/aws-samples/aws-mainframe-modernization-carddemo/blob/main/app/cbl/CBSTM03B.CBL](https://github.com/aws-samples/aws-mainframe-modernization-carddemo/blob/main/app/cbl/CBSTM03B.CBL)

```

98
99      LINKAGE SECTION.
100     01  LK-M03B-AREA.
101         05  LK-M03B-DD          PIC X(08).
102         05  LK-M03B-OPER       PIC X(01).
103             88  M03B-OPEN      VALUE '0'.
104             88  M03B-CLOSE     VALUE 'C'.
105             88  M03B-READ      VALUE 'R'.
106             88  M03B-READ-K    VALUE 'K'.
107             88  M03B-WRITE     VALUE 'W'.
108             88  M03B-REWRITE   VALUE 'Z'.
109         05  LK-M03B-RC          PIC X(02).
110         05  LK-M03B-KEY         PIC X(25).
111         05  LK-M03B-KEY-LN     PIC S9(4).
112         05  LK-M03B-FLDT       PIC X(1000).
113
114     PROCEDURE DIVISION USING LK-M03B-AREA.
115

```

Quindi il programma CBSTM03B accetta un singolo Record come parametro (un array di dimensione 1). Questo è ciò che `CallBuilder` sta costruendo, usando il concatenamento dei metodi `byReference ()` e `getArguments ()`.

La classe API `CallBuilder` fluent dispone di diversi metodi per compilare l'array di argomenti da passare a un chiamante:

- `asPointer (RecordAdaptable)`: aggiunge un argomento di tipo puntatore, per riferimento. Il puntatore rappresenta l'indirizzo di una struttura dati di destinazione.
- `byReference (RecordAdaptable)`: aggiunge un argomento per riferimento. Il chiamante vedrà le modifiche eseguite dal chiamante.
- `byReference (RecordAdaptable...)`: variante `varargs` del metodo precedente.
- `byValue (Object)`: aggiunge un argomento, trasformato in `aRecord`, per valore. Il chiamante non vedrà le modifiche eseguite dal chiamante.
- `byValue (RecordAdaptable)`: uguale al metodo precedente, ma l'argomento è direttamente disponibile come `RecordAdaptable`.

- `byValueWithBounds (Object, int, int)`: aggiunge un argomento, trasformato in `aRecord`, estraendo la parte dell'array di byte definita dai limiti specificati, per valore.

Infine, il metodo `getArguments` raccoglierà tutti gli argomenti aggiunti e li restituirà come matrice di `Record`

#### Note

È responsabilità del chiamante assicurarsi che l'array di argomenti abbia la dimensione richiesta, che gli elementi siano ordinati correttamente e compatibili, in termini di layout di memoria, con i layout previsti per gli elementi di collegamento.

### Script che chiamano programmi

La chiamata ai programmi registrati da groovy script richiede l'utilizzo di un'istanza di classe che implementa l'interfaccia `MainProgramRunner`. Di solito, l'ottenimento di un'istanza del genere si ottiene tramite l'utilizzo di Spring: `ApplicationContext`

```
REPROC.groovy X
1 // Import
2 import com.netfactive.bluage.gapwalk.rt.provider.ScriptRegistry
3 import com.netfactive.bluage.gapwalk.rt.call.MainProgramRunner
4 import com.netfactive.bluage.gapwalk.io.support.FileConfigurationUtils
5 import com.netfactive.bluage.gapwalk.rt.job.support.DefaultJobContext
6 import com.netfactive.bluage.gapwalk.rt.utils.GroovyUtils
7 import com.netfactive.bluage.gapwalk.rt.io.support.FileConfiguration
8 import com.netfactive.bluage.gapwalk.rt.shared.AbendException
9 import com.netfactive.bluage.gapwalk.rt.call.exception.GroovyExecutionException
10 // Variables
11 mpr = applicationContext.getBean("com.netfactive.bluage.gapwalk.rt.call.ExecutionController", MainProgramRunner.class)
12 TreeMap mapTransfo = [:]
```

Dopo che un'interfaccia `MainProgramRunner` è disponibile, utilizzate il metodo `runProgram` per chiamare un programma e passate l'identificatore del programma di destinazione come parametro:

```

REPROC.groovy x
50 //*****
51 /**                                STEPS                                *
52 //*****
53 // STEP PRC001 - PGM - IDCAMS*****
54 def stepPRC001(Object shell, Map params, Map programResults){
55     shell.with {
56         if (checkValidProgramResults(programResults)) {
57             return execStep("PRC001", "IDCAMS", programResults, {
58                 mpr
59                     .withFileConfigurations(new FileConfigurationUtils()
60                         .systemOut("SYSPRINT")
61                         .output("*")
62                         .build()
63                         .bluesam("FILEIN")
64                         .dataset("NULLFILE")
65                         .disposition("SHR")
66                         .build()
67                         .bluesam("FILEOUT")
68                         .dataset("NULLFILE")
69                         .disposition("SHR")
70                         .build()
71                         .fileSystem("SYSIN")
72                         .path("&CNTLLIB(REPROCT)")
73                         .disposition("SHR")
74                         .build()
75                         .getFileConfigurations(fcmmap))
76                     .withParameters(params)
77                     .runProgram("IDCAMS")
78             })
79         }
80     }
81 }

```

Nell'esempio precedente, una fase di processo richiama IDCAMS (programma di utilità per la gestione dei file), che fornisce una mappatura tra le definizioni effettive dei set di dati e i relativi identificatori logici.

Quando si tratta di set di dati, i programmi legacy utilizzano principalmente nomi logici per identificare i set di dati. Quando il programma viene richiamato da uno script, lo script deve mappare i nomi logici con set di dati fisici effettivi. Questi set di dati potrebbero trovarsi sul filesystem, in un archivio Blusam o persino definiti da un flusso in linea, dalla concatenazione di diversi set di dati o dalla generazione di un GDG.

Usa il `withFileConfiguration` metodo per creare una mappa da logica a fisica dei set di dati e renderla disponibile al programma chiamato.



## Scrivi il tuo programma

Scrivere il proprio programma per script o altri programmi modernizzati da richiamare è un'attività comune. In genere, nei progetti di modernizzazione, si scrivono i propri programmi quando un programma eseguibile legacy viene scritto in un linguaggio non supportato dal processo di modernizzazione, oppure i sorgenti sono andati perduti (sì, ciò può succedere) oppure il programma è un'utilità i cui sorgenti non sono disponibili.

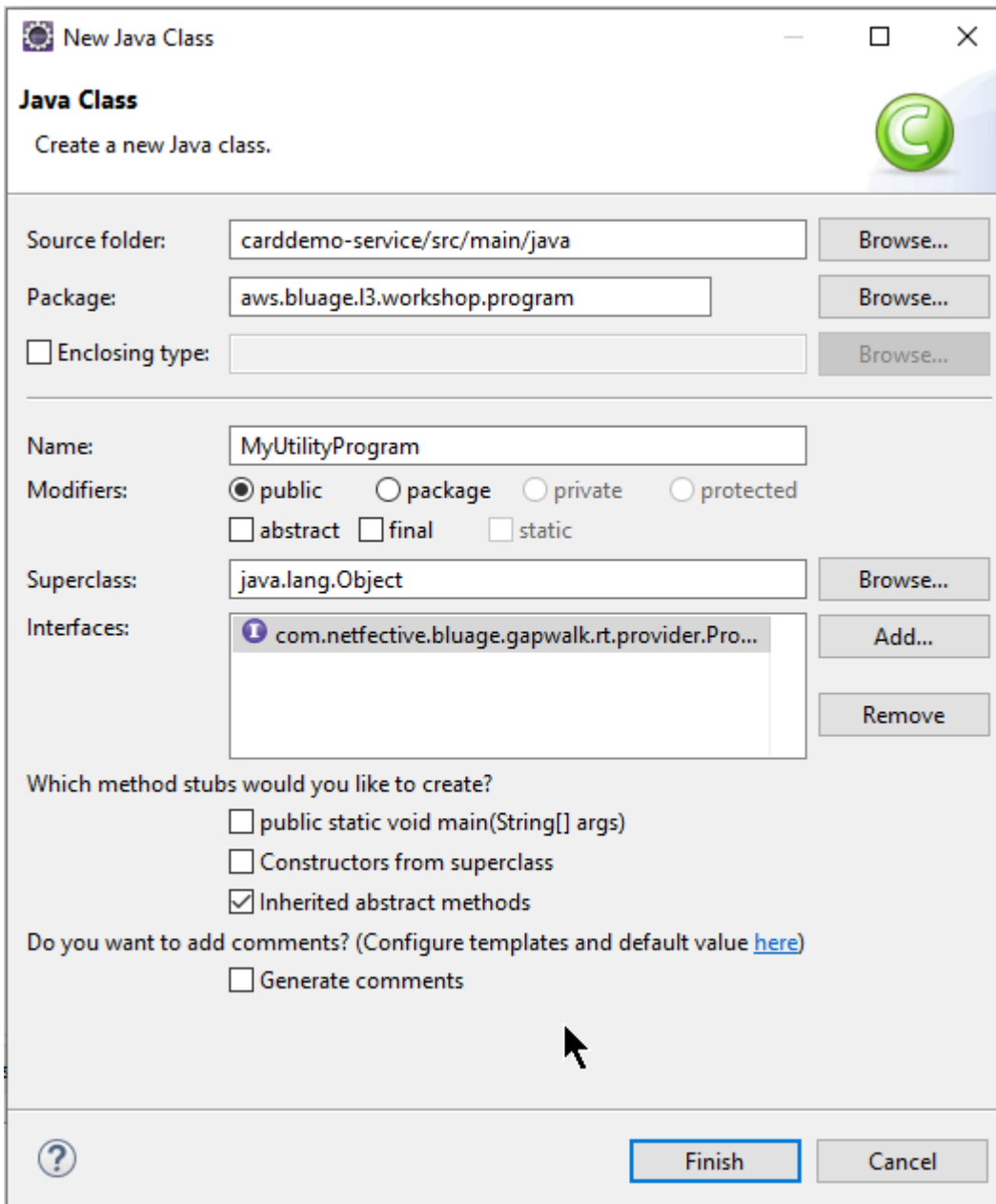
In tal caso, potresti dover scrivere il programma mancante, in java, da solo (supponendo che tu abbia una conoscenza sufficiente di quale dovrebbe essere il comportamento previsto del programma, del layout di memoria degli eventuali argomenti del programma e così via). Il programma java deve essere conforme ai meccanismi di programma descritti in questo documento, in modo che altri programmi e script possano eseguirlo.

Per assicurarsi che il programma sia utilizzabile, è necessario completare due passaggi obbligatori:

- Scrivete una classe che implementi l'`Programinterfaccia` correttamente, in modo che possa essere registrata e chiamata.
- Assicuratevi che il programma sia registrato correttamente, in modo che sia visibile da altri programmi/script.

### Scrittura dell'implementazione del programma

Usa il tuo IDE per creare una nuova classe java che implementa l'`Programinterfaccia`:



L'immagine seguente mostra l'IDE di Eclipse, che si occupa di creare tutti i metodi obbligatori da implementare:

```

MyUtilityProgram.java x
1 package aws.bluage.l3.workshop.program;
2
3 import java.util.Set;
10
11 public class MyUtilityProgram implements Program {
12
13     @Override
14     public ConfigurableApplicationContext getSpringApplication() {
15         // TODO Auto-generated method stub
16         return null;
17     }
18
19     @Override
20     public Set<String> getProgramIdentifiers() {
21         // TODO Auto-generated method stub
22         return null;
23     }
24
25     @Override
26     public Context getContext() {
27         // TODO Auto-generated method stub
28         return null;
29     }
30
31     @Override
32     public void run(ExecutionController ctrl) {
33         // TODO Auto-generated method stub
34
35     }
36
37 }
38

```

## Integrazione primaverile

Innanzitutto, la classe deve essere dichiarata come componente Spring. Annota la classe con l'@Component annotazione:

```

import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.stereotype.Component;

import com.netfactive.bluage.gapwalk.rt.call.ExecutionController;
import com.netfactive.bluage.gapwalk.rt.context.Context;
import com.netfactive.bluage.gapwalk.rt.provider.Program;

import aws.bluage.l3.workshop.SpringBootLauncher;

@Component
public class MyUtilityProgram implements Program {

```

Quindi, implementa correttamente i metodi richiesti. Nel contesto di questo esempio, abbiamo aggiunto MyUtilityProgram al pacchetto che contiene già tutti i programmi modernizzati. Tale

posizionamento consente al programma di utilizzare l'applicazione Springboot esistente per fornire quanto necessario `ConfigurableApplicationContext` per l'implementazione del metodo: `getSpringApplication`

```
public class MyUtilityProgram implements Program {
    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        return SpringBootLauncher.getCac();
    }
}
```

Potresti scegliere una posizione diversa per il tuo programma. Ad esempio, è possibile individuare il programma specificato in un altro progetto di servizio dedicato. Assicurati che il progetto di servizio specificato abbia la sua applicazione Springboot, che consenta di recuperare `ApplicationContext` (che dovrebbe essere un). `ConfigurableApplicationContext`

### Dare un'identità al programma

Per essere richiamabile da altri programmi e script, al programma deve essere assegnato almeno un identificatore, che non deve entrare in conflitto con nessun altro programma registrato esistente nel sistema. La scelta dell'identificatore potrebbe essere determinata dalla necessità di includere un programma precedente sostitutivo; in tal caso, sarà necessario utilizzare l'identificatore previsto, come indicato nelle occorrenze CALL presenti nei programmi legacy. La maggior parte degli identificatori di programma ha una lunghezza di 8 caratteri nei sistemi legacy.

La creazione di un set non modificabile di identificatori nel programma è un modo per farlo. L'esempio seguente mostra la scelta di «MYUTILPG» come identificatore singolo:

```
@Component
public class MyUtilityProgram implements Program {
    /**
     * Unique identifiers for the contained program.
     */
    private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("MYUTILPG").collect(Collectors.toSet()));

    public ConfigurableApplicationContext getSpringApplication() {
        I

    @Override
    public Set<String> getProgramIdentifiers() {
        return programIdentifiers;
    }
}
```

### Associa il programma a un contesto

Il programma necessita di un'`RuntimeContext`istanza complementare. Per i programmi modernizzati, AWS Blu Age genera automaticamente il contesto complementare, utilizzando le strutture di dati che fanno parte del programma legacy.

Se state scrivendo il vostro programma, dovete scrivere anche il contesto complementare.

Facendo riferimento a [Classi relative al programma](#), puoi vedere che un programma richiede almeno due classi complementari:

- una classe di configurazione.
- una classe di contesto che utilizza la configurazione.

Se il programma di utilità utilizza una struttura di dati aggiuntiva, anch'essa dovrebbe essere scritta e utilizzata dal contesto.

Queste classi dovrebbero trovarsi in un pacchetto che fa parte di una gerarchia di pacchetti che verrà analizzata all'avvio dell'applicazione, per assicurarsi che il componente di contesto e la configurazione vengano gestiti dal framework Spring.

Scriviamo una configurazione e un contesto minimi, nel *base package.myutilityprogram.business.context* pacchetto, appena creato nel progetto entities:

```
▼ aws.bluage.I3.workshop.csutldtc.business.model
  > FeedbackCode.java
  > LsDate.java
  > LsDateFormat.java
  > LsResult.java
  > OutputLillian.java
  > WsDateFormat.java
  > WsDateToTest.java
  > WsMessage.java
▼ aws.bluage.I3.workshop.myutilityprogram.business.context
  > MyUtilityProgramConfiguration.java
  > MyUtilityProgramContext.java
```

Ecco il contenuto della configurazione. Utilizza una build di configurazione simile ad altri programmi modernizzati nelle vicinanze. Probabilmente dovrai personalizzarlo per le tue esigenze specifiche.

```
MyUtilityProgramConfiguration.java X
1 package aws.bluage.l3.workshop.myutilityprogram.business.context;
2
3 import java.nio.charset.Charset;
4
5 import org.springframework.context.annotation.Bean;
6 import org.springframework.context.annotation.Lazy;
7
8 import com.netfactive.bluage.gapwalk.datasimplifier.configuration.Configuration;
9 import com.netfactive.bluage.gapwalk.datasimplifier.configuration.ConfigurationBuilder;
10
11 /**
12  * Creates Datasimplifier configuration for the MyUtilityProgram context.
13  */
14 @org.springframework.context.annotation.Configuration
15 @Lazy
16 public class MyUtilityProgramConfiguration {
17
18     @Bean(name = "MyUtilityProgramContextConfiguration")
19     public Configuration configuration() {
20         return new ConfigurationBuilder()
21             .encoding(Charset.forName("CP1047"))
22             .humanReadableEncoding(Charset.forName("ISO-8859-15"))
23             .initDefaultByte(0)
24             .build();
25     }
26 }
27
```

#### Note:

- La convenzione di denominazione generale è ProgramNameConfigurazione.
- Deve utilizzare le annotazioni `@org.springframework.context.annotation.Configuration` e `@Lazy`.
- Il nome del bean di solito segue la convenzione, ma questo non è obbligatorio.  
ProgramNameContextConfiguration Assicurati di evitare le collisioni tra i nomi dei bean all'interno del progetto.
- Il singolo metodo da implementare deve restituire un `Configuration` oggetto. Usa l'API `ConfigurationBuilder` fluent per aiutarti a crearne una.

E il contesto associato:

```

MyUtilityProgramContext.java X
2
3 import org.springframework.beans.factory.annotation.Qualifier;
4 import org.springframework.context.annotation.Lazy;
5 import org.springframework.context.annotation.Scope;
6 import org.springframework.stereotype.Component;
7
8 import com.netflective.bluage.gapwalk.datasimplifier.configuration.Configuration;
9 import com.netflective.bluage.gapwalk.rt.context.RuntimeContext;
10
11 @Component("aws.bluage.13.workshop.myutilityprogram.business.context.MyUtilityProgramContext")
12 @Lazy
13 @Scope("prototype")
14 public class MyUtilityProgramContext extends RuntimeContext{
15
16     protected MyUtilityProgramContext(@Qualifier("MyUtilityProgramContextConfiguration") Configuration configuration) {
17         super(configuration);
18     }
19
20     @Override
21     public void cleanUp() {
22         // TODO implement clean-up of associated data structures if any
23     }
24
25     @Override
26     protected void doReset() {
27         // TODO implement reset of associated data structures if any
28     }
29
30 }
31

```

## Note

- La classe context dovrebbe estendere un'implementazione di Context interfaccia esistente (RuntimeContextoJicsRuntimeContext, che è migliorata RuntimeContext con elementi specifici JICS).
- La convenzione di denominazione generale è Context. ProgramName
- È necessario dichiararlo come componente Prototype e utilizzare l'annotazione @Lazy.
- Il costruttore fa riferimento alla configurazione associata, utilizzando l'annotazione @Qualifier per indirizzare la classe di configurazione corretta.
- Se il programma di utilità utilizza alcune strutture di dati aggiuntive, queste dovrebbero essere:
  - scritto e aggiunto al *base package.business.model* pacchetto.
  - referenziato nel contesto. Dai un'occhiata ad altre classi di contesto esistenti per vedere come fare riferimento alle classi di strutture di dati e adattare i metodi di contesto (constructor/cleanup/reset) secondo necessità.

Ora che è disponibile un contesto dedicato, lascia che il nuovo programma lo utilizzi:

```

MyUtilityProgram.java ×
10
19 import aws.bluage.l3.workshop.SpringBootLauncher;
20
21 @Component
22 @Import({
23     aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramConfiguration.class,
24     aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class
25 })
26 public class MyUtilityProgram implements Program {
27
28     @Autowired
29     BeanFactory beanFactory;
30
31     /**
32      * Unique identifiers for the contained program.
33      */
34     private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("MYUTILPG").collect(Collectors.toSet()));
35
36     private static final String programIdentifier = "MYUTILPG";
37
38     @Override
39     public ConfigurableApplicationContext getSpringApplication() {
40         return SpringBootLauncher.getCac();
41     }
42
43     @Override
44     public Set<String> getProgramIdentifiers() {
45         return programIdentifiers;
46     }
47
48     /**
49      * {@inheritDoc}
50      */
51     @Override
52     public String getProgramMainIdentifier() {
53         return programIdentifier;
54     }
55
56
57     @Override
58     public Context getContext() {
59         return ProgramContextStore.getOrCreate(
60             getProgramMainIdentifier(),
61             aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class,
62             beanFactory);
63     }
64

```

## Note:

- Il metodo getContext deve essere implementato rigorosamente come mostrato, utilizzando una delega al getOrCreate metodo della ProgramContextStore classe e il comando automatico Spring. BeanFactory Un singolo identificatore di programma viene utilizzato per memorizzare il contesto del programma in ProgramContextStore; questo identificatore viene indicato come «identificatore principale del programma».
- È necessario fare riferimento alla configurazione complementare e alle classi di contesto utilizzando l'annotazione spring. @Import

## Implementazione della logica aziendale

Una volta completata la struttura del programma, implementate la logica di business per il nuovo programma di utilità.



Fatelo con il `run` metodo del programma. Questo metodo verrà eseguito ogni volta che il programma viene chiamato, da un altro programma o da uno script.

Buona codifica!

## Gestione della registrazione al programma

Infine, assicurati che il nuovo programma sia registrato correttamente in `ProgramRegistry`. Se avete aggiunto il nuovo programma al pacchetto che già contiene altri programmi, non c'è altro da fare. Il nuovo programma viene prelevato e registrato con tutti i programmi adiacenti all'avvio dell'applicazione.

Se hai scelto un'altra posizione per il programma, devi assicurarti che il programma sia registrato correttamente all'avvio di Tomcat. Per qualche ispirazione su come farlo, date un'occhiata al metodo di inizializzazione delle `SpringbootLauncher` classi generate nei progetti di servizio (vedi [Contenuto del progetto di servizio](#)).

Controlla i log di avvio di Tomcat. Ogni registrazione al programma viene registrata. Se il programma è stato registrato correttamente, troverai la voce di registro corrispondente.

Quando sei sicuro che il tuo programma sia registrato correttamente, puoi iniziare a ripetere la codifica della logica aziendale.

## Mappature dei nomi complete

Questa sezione contiene elenchi dei mapping dei nomi completi di AWS Blu Age e di terze parti da utilizzare nelle applicazioni modernizzate.

### AWS Mappature dei nomi completamente qualificate di Blu Age

Nome breve	Nome completamente qualificato
<code>CallBuilder</code>	<code>com.netfactive.bluage.gapwalk.runtime.statements.CallBuilder</code>
<code>Configuration</code>	<code>com.netfactive.bluage.gapwalk.datasimplifier.configuration.Configuration</code>

Nome breve	Nome completamente qualificato
ConfigurationBuilder	com.netfective.bluage.gapwalk.datasimplifier.configuration.ConfigurationBuilder
ExecutionController	com.netfective.bluage.gapwalk.rt.call.ExecutionController
ExecutionControllerImpl	com.netfective.bluage.gapwalk.rt.call.internal.Executi onControllerImpl
File	com.netfective.bluage.gapwalk.rt.io.File
MainProgramRunner	com.netfective.bluage.gapwalk.rt.call.MainProgramRunner
Program	com.netfective.bluage.gapwalk.rt.provider.Program
ProgramContextStore	com.netfective.bluage.gapwalk.rt.context.ProgramContex tStore
ProgramRegistry	com.netfective.bluage.gapwalk.rt.provider.ProgramRegistry
Record	com.netfective.bluage.gapwalk.datasimplifier.data.Record
RecordEntity	com.netfective.bluage.gapwalk.datasimplifier.entity.Re cordEntity
RuntimeContext	com.netfective.bluage.gapwalk.rt.context.RuntimeContext

Nome breve	Nome completamente qualificato
SimpleStateMachineController	com.netfective.bluage.gapwalk.rtm.statemachine.SimpleStateMachineController
StateMachineController	com.netfective.bluage.gapwalk.rtm.statemachine.StateMachineController
StateMachineRunner	com.netfective.bluage.gapwalk.rtm.statemachine.StateMachineRunner

### Mappature dei nomi completamente qualificate di terze parti

Nome breve	Nome completamente qualificato
@Autowired	org.springframework.beans.factory.annotation.Autowired
@Bean	org.springframework.context.annotation.Bean
BeanFactory	org.springframework.beans.factory.BeanFactory
@Component	org.springframework.stereotype.Component
ConfigurableApplicationContext	org.springframework.context.ConfigurableApplicationContext
@Import	org.springframework.context.annotation.Import
@Lazy	org.springframework.context.annotation.Lazy

## Semplificatore di dati

Sui sistemi mainframe e midrange (indicati nell'argomento seguente come sistemi «legacy»), i linguaggi di programmazione più utilizzati come COBOL, PL/I o RPG forniscono un accesso di basso livello alla memoria. Questo accesso si concentra sul layout della memoria a cui si accede tramite tipi nativi, ad esempio suddivisi in zone, compressi o alfanumerici, possibilmente anche aggregati tramite gruppi o array.

In un determinato programma coesiste una combinazione di accessi a un determinato pezzo di memoria, tramite entrambi i campi digitati e come accesso diretto ai byte (memoria grezza). Ad esempio, i programmi COBOL passeranno gli argomenti ai chiamanti come set di byte contigui (LINKAGE) o leggeranno e scriveranno i dati dai file nello stesso modo (record), interpretando tali intervalli di memoria con campi tipizzati organizzati in quaderni.

Tali combinazioni di accesso grezzo e strutturato alla memoria, il ricorso a un layout di memoria preciso a livello di byte e tipi legacy, ad esempio suddivisi in zone o compressi, sono funzionalità che non sono né nativamente né facilmente disponibili nell'ambiente di programmazione Java.

Come parte della soluzione AWS Blu Age per modernizzare i programmi legacy in Java, la libreria Data Simplifier fornisce tali costrutti ai programmi Java modernizzati e li espone in un modo il più familiare possibile agli sviluppatori Java (getters/setter, array di byte, basati su classi). È una dipendenza fondamentale del codice Java modernizzato generato da tali programmi.

Per semplicità, la maggior parte delle spiegazioni seguenti si basa su costrutti COBOL, ma è possibile utilizzare la stessa API per la modernizzazione del layout dei dati PL1 e RPG, poiché la maggior parte dei concetti è simile.

### Argomenti

- [Classi principali](#)
- [Associazione e accesso ai dati](#)
- [FQN dei tipi Java discussi](#)

### Classi principali

Per facilitare la lettura, questo documento utilizza i nomi brevi Java delle interfacce e delle classi dell'API AWS Blu Age. Per ulteriori informazioni, consulta [FQN dei tipi Java discussi](#).

## Rappresentazione della memoria a basso livello

Al livello più basso, la memoria (un intervallo contiguo di byte accessibile in modo rapido e casuale) è rappresentata dall'interfaccia. Record Questa interfaccia è essenzialmente un'astrazione di un array di byte di dimensione fissa. In quanto tale, fornisce setter e getter in grado di accedere o modificare i byte sottostanti.

## Rappresentazione strutturata dei dati

Per rappresentare dati strutturati, come «01 elementi di dati» o «01 quaderni», come si trovano in COBOL DATA DIVISION, vengono utilizzate le sottoclassi della `RecordEntity` classe. Normalmente non sono scritti a mano, ma generati dagli strumenti di modernizzazione AWS Blu Age a partire dai corrispondenti costrutti legacy. È comunque utile conoscere la loro struttura principale e le API, in modo da poter capire come vengono utilizzate dal codice di un programma modernizzato. Nel caso di COBOL, quel codice è Java generato dalla relativa PROCEDURE DIVISION.

Il codice generato rappresenta ogni «elemento di dati 01" con una `RecordEntity` sottoclasse; ogni campo elementare o aggregato che lo compone è rappresentato come un campo Java privato, organizzato come un albero (ogni elemento ha un elemento principale, tranne quello principale).

A scopo illustrativo, ecco un esempio di elemento di dati COBOL, seguito dal corrispondente codice generato da AWS Blu Age che lo modernizza:

```
01 TST2.  
 02 FILLER PIC X(4).  
 02 F1      PIC 9(2) VALUE 42.  
 02 FILLER PIC X.  
 02        PIC 9(3) VALUE 123.  
 02 F2      PIC X VALUE 'A'.
```

```
public class Tst2 extends RecordEntity {  
  
    private final Group root = new Group(getData()).named("TST2");  
    private final Filler filler = new Filler(root,new AlphanumericType(4));  
    private final Elementary f1 = new Elementary(root,new ZonedType(2, 0, false),new  
BigDecimal("42")).named("F1");  
    private final Filler filler1 = new Filler(root,new AlphanumericType(1));  
    private final Filler filler2 = new Filler(root,new ZonedType(3, 0, false),new  
BigDecimal("123"));  
    private final Elementary f2 = new Elementary(root,new  
AlphanumericType(1),"A").named("F2");  
}
```

```
/**
 * Instantiate a new Tst2 with a default record.
 * @param configuration the configuration
 */
public Tst2(Configuration configuration) {
    super(configuration);
    setupRoot(root);
}
/**
 * Instantiate a new Tst2 bound to the provided record.
 * @param configuration the configuration
 * @param record the existing record to bind
 */
public Tst2(Configuration configuration, RecordAdaptable record) {
    super(configuration);
    setupRoot(root, record);
}

/**
 * Gets the reference for attribute f1.
 * @return the f1 attribute reference
 */
public ElementaryRangeReference getF1Reference() {
    return f1.getReference();
}

/* *
 * Getter for f1 attribute.
 * @return f1 attribute
 */
public int getF1() {
    return f1.getValue();
}

/**
 * Setter for f1 attribute.
 * @param f1 the new value of f1
 */
public void setF1(int f1) {
    this.f1.setValue(f1);
}
/**
 * Gets the reference for attribute f2.
```

```

    * @return the f2 attribute reference
    */
    public ElementaryRangeReference getF2Reference() {
        return f2.getReference();
    }

    /**
     * Getter for f2 attribute.
     * @return f2 attribute
     */
    public String getF2() {
        return f2.getValue();
    }

    /**
     * Setter for f2 attribute.
     * @param f2 the new value of f2
     */
    public void setF2(String f2) {
        this.f2.setValue(f2);
    }
}

```

## Campi elementari

I campi di classe `Elementary` (`oFiller`, se senza nome) rappresentano una «foglia» della struttura dei dati preesistente. Sono associati a un intervallo contiguo di byte sottostanti («intervallo») e di solito hanno un tipo (possibilmente parametrizzato) che esprime come interpretare e modificare tali byte (rispettivamente «decodificando» e «codificando» un valore da/verso un array di byte).

Tutti i tipi elementari sono `RangeType` sottoclassi di. I tipi più comuni sono:

Tipo COBOL	Tipo di semplificatore di dati
PIC X(n)	<code>AlphanumericType</code>
PIC 9(n)	<code>ZonedType</code>
PIC 9(n) COMP-3	<code>PackedType</code>
PIC 9(n) COMP-5	<code>BinaryType</code>

## Campi aggregati

I campi aggregati organizzano il layout di memoria del loro contenuto (altri aggregati o campi elementari). Essi stessi non hanno un tipo elementare.

Groupi campi rappresentano campi contigui in memoria. Ciascuno dei campi contenuti è disposto nello stesso ordine in memoria, il primo campo è in offset 0 rispetto alla posizione del campo del gruppo in memoria, il secondo campo è in  $0 + (\text{size in bytes of first field})$  offset, ecc. Vengono utilizzati per rappresentare sequenze di campi COBOL nello stesso campo contenitore.

Unioni campi rappresentano più campi che accedono alla stessa memoria. Ciascuno dei campi contenuti è disposto in modo offset 0 rispetto alla posizione del campo unionale in memoria. Vengono ad esempio utilizzati per rappresentare il costrutto COBOL «REDEFINES» (i primi figli dell'Unione sono l'elemento di dati ridefinito, i secondi figli ne sono la prima ridefinizione, ecc.).

I campi array (sottoclassi di `Repetition`) rappresentano la ripetizione, in memoria, del layout del loro campo figlio (sia esso un aggregato stesso o un elemento elementare). Presentano un determinato numero di tali layout secondari in memoria, ognuno dei quali è in posizione sfalsata. `index * (size in bytes of child)` Sono usati per rappresentare i costrutti COBOL «OCCURS».

## Primitive

In alcuni casi di modernizzazione, i «primitivi» possono essere utilizzati anche per presentare elementi di dati «root» indipendenti. Sono molto simili nell'uso `RecordEntity` ma non provengono da esso, né si basano su codice generato. Vengono invece forniti direttamente dal runtime AWS Blu Age come sottoclassi dell'`Primitive` interfaccia. Esempi di tali classi fornite sono `Alphanumeric` o `ZonedDecimal`.

## Associazione e accesso ai dati

L'associazione tra dati strutturati e dati sottostanti può essere effettuata in diversi modi.

Un'interfaccia importante a questo scopo è quella `RecordAdaptable` che viene utilizzata per ottenere `Record` una «visualizzazione scrivibile» dei dati `RecordAdaptable` sottostanti. Come vedremo di seguito, vengono `RecordAdaptable` implementate più classi. Reciprocamente, le API AWS Blu Age e il codice che manipola la memoria di basso livello (come gli argomenti dei programmi, i record di I/O dei file, la commarea CICS, la memoria allocata...) spesso si aspettano un `handle` per quella memoria. `RecordAdaptable`

Nel caso della modernizzazione COBOL, la maggior parte degli elementi di dati è associata alla memoria, che verrà fissata durante la durata dell'esecuzione del programma corrispondente. A



tal fine, `RecordEntity` le sottoclassi vengono istanziate una sola volta in un oggetto principale generato (il programma `Context`) e si occuperanno di istanziare le relative sottoclassi, in base alla `Record` dimensione dei byte. `RecordEntity`

In altri casi COBOL, come l'associazione di elementi `LINKAGE` agli argomenti del programma o la modernizzazione del costrutto `SET ADDRESS OF`, un'istanza deve essere associata a un oggetto fornito. `RecordEntity RecordAdaptable` A tal fine, esistono due meccanismi:

- se l'`RecordEntity` istanza esiste già, il `RecordEntity.bind(RecordAdaptable)` metodo (ereditato da `Bindable`) può essere usato per fare in modo che questa istanza «punti» a questo `RecordAdaptable`. Qualsiasi getter o setter chiamato su `RecordEntity` verrà quindi supportato (lettura o scrittura di byte) dai byte sottostanti. `RecordAdaptable`
- se deve essere istanziato, `RecordEntity` è disponibile un costruttore generato che accetta a. `RecordAdaptable`

Al contrario, è possibile accedere ai dati `Record` attualmente associati ai dati strutturati. Per questo, `RecordEntity` implementa `RecordAdaptable`, quindi `getRecord()` può essere chiamato su qualsiasi istanza del genere.

Infine, molti verbi COBOL o CICS richiedono l'accesso a un singolo campo, per scopi di lettura o scrittura. La `RangeReference` classe viene utilizzata per rappresentare tale accesso. Le sue istanze possono essere ottenute dai `getXXXReference()` metodi `RecordEntity` generati (ovvero `XXX` il campo a cui si accede) e passate ai metodi di runtime. `RangeReference` viene in genere utilizzato per accedere a interi `RecordEntity` o `Group`, mentre la sua sottoclasse `ElementaryRangeReference` rappresenta gli accessi ai campi. `Elementary`

Nota che la maggior parte delle osservazioni precedenti si applica alle `Primitive` sottoclassi, poiché si sforzano di implementare un comportamento simile a quello fornito dal runtime AWS Blu Age (anziché dal codice generato). `RecordEntity` A tal fine, tutte le sottoclassi di `Primitive` implementano `RecordAdaptable` `ElementaryRangeReference` e `Bindable` interfacce in modo da essere utilizzabili al posto delle `RecordEntity` sottoclassi e dei campi elementari.

## FQN dei tipi Java discussi

La tabella seguente mostra i nomi completi dei tipi Java discussi in questa sezione.

Nome breve	Nome completamente qualificato
Alphanumeric	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.Alphanumeric</code>
AlphanumericType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType</code>
BinaryType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.BinaryType</code>
Bindable	<code>com.netfective.bluage.gapwalk.datasimplifier.data.Bindable</code>
Elementary	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary</code>
ElementaryRangeReference	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference</code>
Filler	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Filler</code>
Group	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group</code>
PackedType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.PackedType</code>

Nome breve	Nome completamente qualificato
Primitive	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.Primitive</code>
RangeReference	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference</code>
RangeType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.RangeType</code>
Record	<code>com.netfective.bluage.gapwalk.datasimplifier.data.Record</code>
RecordAdaptable	<code>com.netfective.bluage.gapwalk.datasimplifier.data.RecordAdaptable</code>
RecordEntity	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity</code>
Repetition	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Repetition</code>
Union	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Union</code>
ZonedDecimal	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.ZonedDecimal</code>

Nome breve	Nome completamente qualificato
ZonedType	com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType

## AWS File di configurazione e configurazione di Blu Age Runtime

Il framework Velocity e il codice client sono applicazioni Web che utilizzano il framework [Spring](#) Boot. Sfrutta le funzionalità di Spring per fornire la configurazione, con diverse posizioni possibili e regole di precedenza. Esistono anche regole di precedenza simili per fornire molti altri file, come groovy scripts, sql, ecc.

Il framework Velocity contiene anche applicazioni web opzionali aggiuntive, che possono essere attivate se necessario.

### Argomenti

- [Nozioni di base sulla configurazione delle applicazioni](#)
- [Precedenza delle applicazioni](#)
- [JNDI per database](#)
- [Usare AWS i segreti](#)
- [Altri file \(groovy, sql, ecc.\)](#)
- [Applicazione web aggiuntiva](#)
- [Abilitazione delle proprietà](#)
- [Configurare l'autenticazione per le applicazioni Gapwalk](#)

## Nozioni di base sulla configurazione delle applicazioni

Il modo predefinito per gestire la configurazione delle applicazioni consiste nell'uso di file YAML dedicati da fornire nella cartella del server delle applicazioni. config Esistono due file di configurazione YAML principali:

- application-main.yaml
- application-*profile*.yaml (dove *profile* il valore è impostato durante la generazione dell'applicazione).

Il primo file configura il framework, ad esempio `Gapwalk-application.war`, mentre il secondo riguarda opzioni aggiuntive specifiche per l'applicazione client. Funziona con l'uso di profili Spring: l'applicazione Gapwalk utilizza il `main` profilo, mentre l'applicazione client utilizza il profilo *profile*.

L'esempio seguente mostra un tipico file YAML principale.

```
#####
#### JICS datasource configuration ####
#####
datasource:
  jicsDs:
    driver-class-name : org.postgresql.Driver
    url: jdbc:postgresql://localhost/jics
    username: jics
    password: jics
    type : org.postgresql.ds.PGSimpleDataSource

#####
#### Embedded Bluesam datasource configuration ####
#####
bluesamDs :
  driver-class-name : org.postgresql.Driver
  url : jdbc:postgresql://localhost/bluesam
  username : bluesam
  password : bluesam
  type : org.postgresql.ds.PGSimpleDataSource

#####
#### Embedded Bluesam configuration ####
#####
bluesam :
  remote : false
  cache : ehcache
  persistence : pgsq #pgsql, mssql, xodus...
  ehcache:
    resource-pool:
      size: 4GB
  write-behind:
```

L'esempio seguente mostra un tipico file YAML del client.

```
# Logback context logger integration.
logging.config : classpath:logback-XXXXXXXXXX.xml
# Limits Spring logger output.
logging.level.org.springframework.beans.factory.support.DefaultListableBeanFactory : WARN
logging.level.org.springframework.statemachine : WARN
# If the datasource support mode is not static-xa, spring JTA transactions autoconfiguration must me disabled
spring.jta.enabled : false

spring:
  aws:
    client:
      datasources:
        names: primary
        primary:
          secret: arn:aws:secretsmanager:XXXXXXXXXX

spring.jta.atomikos.datasource.primary.unique-resource-name: primary
spring.jta.atomikos.datasource.primary.xa-data-source-class-name: org.postgresql.xa.PGXADatasource
spring.jta.atomikos.datasource.primary.maxPoolSize: 20
spring.jta.atomikos.datasource.primary.autoCommit: false
```

Per informazioni sul contenuto dei file YAML, vedere. [Abilitazione delle proprietà](#)

## Precedenza delle applicazioni

Per questi file di configurazione, si applicano le regole di precedenza Spring. In particolare:

- Il file `application-main` YAML appare nel file di guerra principale di Gapwalk con i valori predefiniti e quello nella cartella lo sostituisce. `config`
- Lo stesso dovrebbe essere fatto per la configurazione dell'applicazione `client`
- È possibile passare parametri aggiuntivi sulla riga di comando al momento dell'avvio del server. Avrebbero la precedenza su quelli YAML.

Per ulteriori informazioni, consulta la documentazione [ufficiale di Spring](#) Boot.

## JNDI per database

La configurazione del database potrebbe essere fornita con JNDI nel file `context.xml` in Tomcat. Qualsiasi configurazione di questo tipo avrebbe la precedenza su quella YAML. Ma fai attenzione che l'utilizzo di questa opzione non ti consentirà di racchiudere le tue credenziali in un gestore segreto (vedi sotto).

L'esempio seguente mostra configurazioni di esempio per JICS e database. BluSam

```
<Resource auth="Container" driverClassName="org.postgresql.Driver" initialSize="0"
  maxIdle="5"
```

```
maxOpenPreparedStatements="-1" maxTotal="10" maxWaitMillis="-1" name="jdbc/jics"  
poolPreparedStatements="true" testOnBorrow="false" type="javax.sql.DataSource"  
url="jdbc:postgresql://XXXX.rds.amazonaws.com:5432/XXXX" username="XXXX"  
password="XXXX" />
```

## jdbc/jics

Sarebbe `jdbc/jics` per il database JICS e `jdbc/bluesam` (presta attenzione alla «e») per il database blusam.

```
url="jdbc:postgresql://xxxx.rds.amazonaws.com:5432/XXXX» username="XXXX» password="XXXX»
```

L'URL, il nome utente e la password del database.

## Usare AWS i segreti

Alcune configurazioni di risorse che contengono credenziali possono essere ulteriormente protette utilizzando segreti. AWS L'idea è di archiviare i dati critici in un luogo AWS segreto e avere un riferimento al segreto nella configurazione YAML in modo che il contenuto segreto venga selezionato al volo all'avvio di tomcat.

### Segreti per Aurora

La configurazione del database Aurora (per jics, blusam, customer db, ecc.) utilizzerà il [segreto del database](#) integrato, che popolerà automaticamente tutti i campi pertinenti dal database corrispondente.

#### Note

La `dbname` chiave è facoltativa, a seconda della configurazione del database, entrerà nel segreto o meno. Puoi aggiungerla lì manualmente o fornendo il nome al file YAML.

### Altri segreti

Altri segreti riguardano le risorse con un'unica password (in particolare le cache redis protette da password). In questo caso è necessario utilizzare l'[altro tipo di segreto](#), con una sola chiave. `password`

## Riferimenti YAML ai segreti

`application-main.yaml` Possono fare riferimento all'ARN segreto per varie risorse. Le più importanti sono:

- Credenziali del database JICS con `spring.aws.jics.db.secret`
- JICS TS mette in coda le credenziali Redis con `spring.aws.client.jics.queues.ts.redis.secret`
- Credenziali del database Bluesam con `spring.aws.client.bluesam.db.secret`
- Password della cache Bluesam con `spring.aws.client.bluesam.redis.secret`
- Bluesam blocca la password della cache con `spring.aws.client.bluesam.locks.redis.secret`

L'esempio seguente mostra come dichiarare questi segreti in un file YAML.

```
spring:
  aws:
    client:
      bluesam:
        locks:
          redis:
            secret: arn:aws:secretsmanager:XXXX
      db:
        dbname: bluesam
        secret: arn:aws:secretsmanager:XXXX
      redis:
        secret: arn:aws:secretsmanager:XXXX
    jics:
      queues:
        ts:
          redis:
            secret: arn:aws:secretsmanager:XXXX
    jics:
      db:
        secret: arn:aws:secretsmanager:XXXX
```

`dbname: bluesam`

In questo esempio, il nome del database non è nel segreto e viene invece fornito qui.



Il client `application-profile.yml` può fare riferimento all'ARN segreto per il database del client. Ciò richiede una proprietà aggiuntiva per elencare le fonti di dati, illustrata nell'esempio seguente:

```
spring:
  aws:
    client:
      datasources:
        names: primary,host
        primary:
          secret: arn:aws:secretsmanager:XXXX
        host:
          secret: arn:aws:secretsmanager:XXXX
```

nomi: primario, host

Un esempio con due origini dati client denominate `primary` e `host`, ognuna con il proprio database e le proprie credenziali.

dbname: mydb

In questo esempio, il nome del database «host» non è nel segreto e viene invece fornito qui, mentre per il database «primario» è nel segreto.

Nessuna chiave segreta supportata da XA

- motore (postgres/oracle/db2/mssql)
- port
- dbname
- Schema attuale
- username
- password
- url

Perché postgres solo la chiave `sslMode` segreta valued (`disable/allow/prefer/require/verify-ca/verify-full`) oltre alla proprietà `spring.aws.rds.ssl.cert-path` yml consente la connessione con SSL.

## Le chiavi segrete supportate da XA

Se il database del client utilizza XA, le proprietà xa secondarie sono supportate tramite valori segreti.

- host
- port
- dbname
- Schema corrente
- username
- password
- url
- Connessione SSL (vero/falso)

Tuttavia, per altre proprietà xa (ad esempio `maxPoolSize` o `driverType`), deve essere comunque fornita la normale chiave YAML. `spring.jta.atomikos.datasource.XXXX.unique-resource-name`

Il valore segreto ha la precedenza sulle proprietà YAML.

## Altri file (groovy, sql, ecc.)

Gli altri file utilizzati dal progetto del cliente utilizzano regole di precedenza simili a quelle per la configurazione primaverile. Esempi:

- Gli script Groovy sono `.groovy` file nella cartella o nelle sottocartelle. `scripts`
- Gli script SQL sono `.sql` file contenuti nella cartella o nelle sottocartelle. `sql`
- Gli script daemon sono `.groovy` file contenuti nella cartella o nelle sottocartelle. `daemons`
- Queries Database mapping file sono file `queries-database.mapping` denominati file nelle sottocartelle delle cartelle. `sql`
- I modelli Jasper sono `.jrxml` file contenuti nella cartella o nelle sottocartelle `templates`.
- I cataloghi dei set di dati sono `.json` file contenuti nella cartella. `catalog`
- I file Lnk sono `.json` file contenuti nella cartella. `lnk`

Tutte queste posizioni possono essere sovrascritte tramite una proprietà di sistema o una proprietà `yml` del client.

- Per gli script Groovy: `configuration.scripts`
- Per gli script SQL: `configuration.sql`
- Per gli script Daemon: `configuration.daemons`
- Per il file di mappatura del database Queries: `configuration.databaseMapping`
- Per i modelli Jasper: `configuration.templates`
- Per i cataloghi Dataset: `configuration.catalog`
- Per i file Lnk: `configuration.lnk`

Se la proprietà non viene trovata, i file verranno prelevati dalla posizione predefinita sopra menzionata. La ricerca verrà eseguita prima con la directory di lavoro di tomcat come root e infine nel file war dell'applicazione.

## Applicazione web aggiuntiva

Il framework Velocity contiene applicazioni web aggiuntive nella sua `webapps-extra` cartella. Queste applicazioni non sono servite per impostazione predefinita dal server tomcat.

L'attivazione di queste applicazioni Web dipende dal progetto di modernizzazione e viene effettuata spostando il file war desiderato dalla `webapps-extra` cartella alla cartella `webapps`. Dopodiché, la guerra verrà gestita dal server tomcat al prossimo avvio.

È inoltre possibile aggiungere alcune configurazioni aggiuntive specifiche del progetto in un file di configurazione YAML per ogni guerra aggiuntiva, come viene fatto nel file e spiegato sopra. `application-main.yml` Le guerre aggiuntive sono:

- `gapwalk-utility-pgm.war`: contiene il supporto per i programmi di utilità ZOS e li utilizza `application-utility-pgm.yml` come configurazione.
- `gapwalk-cl-command.war`: contiene il supporto per i programmi di utilità AS/400 e li utilizza `application-cl-command.yml` come configurazione.
- `gapwalk-hierarchical-support.war`: contiene il supporto per le transazioni IMS/MFS e lo utilizza come configurazione `application-jhdb.yml`

## Abilitazione delle proprietà

Nelle applicazioni Spring Boot `application-main.yml` c'è il file di configurazione in cui definiamo diversi tipi di proprietà come la porta di ascolto, la connettività del database e molte altre.

## Argomenti

- [Notazione YML](#)
- [Avvio rapido/Casi d'uso](#)
- [Proprietà disponibili per l'applicazione principale](#)
- [Proprietà disponibili per applicazioni web opzionali](#)

## Notazione YML

Nella seguente documentazione, una proprietà come questa `parent.child1.child2=true` è scritta come segue in formato YAML.

```
parent:
  child1:
    child2: true
```

## Avvio rapido/Casi d'uso

I seguenti casi d'uso mostrano esempi delle chiavi e dei valori applicabili.

- Application-main.yml predefinito

```
----
#### DEFAULT APPLICATION-MAIN.YML FILE      #####
#### SHOWING USEFUL CONFIGURATION ELEMENTS #####
#### SHOULD BE OVERRIDDEN AND EXTERNALIZED  #####

#####
##### Logging configuration #####
#####

logging:
  config: classpath:logback-main.xml
  level.org.springframework.beans.factory.support.DefaultListableBeanFactory : WARN

#####
##### Spring configuration #####
#####

spring:
  quartz:
    auto-startup: false
    scheduler-name: Default
```

```

properties:
  org.quartz.threadPool.threadCount: 1
jta:
  enabled: false
  atomikos.properties.maxTimeout : 600000
  atomikos.properties.default-jta-timeout : 100000
jpa:
# DISABLE OpenEntityManagerInViewInterceptor
  open-in-view: false
  # Fix Postgres JPA Error:
  # Method org.postgresql.jdbc.PgConnection.createClob() is not yet implemented.
  properties.hibernate.temp.use_jdbc_metadata_defaults : false
#####
##### Jics tables configuration #####
#####

  # The dialect should match the jics datasource choice
  database-platform : org.hibernate.dialect.PostgreSQLDialect #
org.hibernate.dialect.PostgreSQLDialect, org.hibernate.dialect.SQLServerDialect

  # those properties can be used to create and initialize jics tables
  automatically.
#   properties:
#     hibernate:
#       globally_quoted_identifiers: true
#       hbm2ddl:
#         import_files_sql_extractor :
org.hibernate.tool.hbm2ddl.MultipleLinesSqlCommandExtractor
#         import_files : file:./setup/initJics.sql
#         auto : create

#####
##### Level 2 cache #####
#####
#       cache:
#         use_second_level_cache: true
#         use_query_cache: true
#         region:
#           factory_class: org.hibernate.cache.ehcache.EhCacheRegionFactory
#     javax:
#       persistence:
#         sharedCache:
#           mode: ENABLE_SELECTIVE
#####

```

```

##### Redis settings #####
#####
session:
  store-type: none #redis

#####
##### JICS datasource configuration #####
#####
datasource:
  jicsDs:
    driver-class-name : org.postgresql.Driver # org.postgresql.Driver,
com.microsoft.sqlserver.jdbc.SQLServerDriver
    url: jdbc:postgresql://localhost/jics # jdbc:postgresql://localhost:5433/jics,
jdbc:sqlserver://localhost\SQLEXPRESS:1434;databasename=jics;
    username: jics
    password: jics
    type : org.postgresql.ds.PGSimpleDataSource #
org.postgresql.ds.PGSimpleDataSource,
com.microsoft.sqlserver.jdbc.SQLServerDataSource

#####
##### Embedded Bluesam datasource configuration #####
#####
bluesamDs :
  driver-class-name : org.postgresql.Driver # org.postgresql.Driver,
com.microsoft.sqlserver.jdbc.SQLServerDriver
  url : jdbc:postgresql://localhost/bluesam # jdbc:postgresql://localhost:5433/
jics, jdbc:sqlserver://localhost\SQLEXPRESS:1434;databasename=jics;
  username : bluesam
  password : bluesam
  type : org.postgresql.ds.PGSimpleDataSource #
org.postgresql.ds.PGSimpleDataSource,
com.microsoft.sqlserver.jdbc.SQLServerDataSource

#####
##### Embedded Bluesam configuration #####
#####
bluesam :
  remote : false
  cache : ehcache
  persistence : pgsq1 #pgsq1, mssql, xodus...
  ehcache:
    resource-pool:
      size: 4GB

```

```

write-behind:
  enabled: true
pgsql :
  dataSource : bluesamDs

#####
##### Jics settings #####
#####
rabbitmq.host: localhost
jics:
  cache: false #redis
  resource-definitions.store-type: jpa # default value: jpa, other possible value:
redis
redis.hostname: 127.0.0.1 # Redis server host.
redis.password: redis # Login password of the redis server.
redis.port: 6379 # Redis server port.
redis.username: # Redis username
redis.mode: standalone # Redis mode. Possible values: standalone, cluster
jics.disableSyncpoint : false
#jics.initList:
#jics.parameters.datform: DDMMYY
#jics.parameters.applid: VELOCITY
#jics.parameters.sysid: CICS
#jics.parameters.eibtrmid: TERM
#jics.parameters.userid: MYUSERID
#jics.parameters.username: MYUSERNAME
#jics.parameters.opid: XXX
#jics.parameters.cwa.length: 0
#jics.parameters.netname: MYNETNAME
#jics.parameters.jobname: MJOBNAME
#jics.parameters.sysname: SYSNAME

#####
##### Jics RunUnitLauncher pool settings #####
#####
#jics.runUnitLauncherPool.enable: false
#jics.runUnitLauncherPool.size: 20
#jics.runUnitLauncherPool.validationInterval: 1000

#####
##### Jhdb settings #####
#####
#jhdb.lterm: LTERMVAL
#jhdb.identificationCardData: SomeIDData

```

```
#####  
##### DateHelper configuration #####  
#####  
#forcedDate: "2013-08-26T12:59:58+01:57"  
  
#####  
##### Sort configuration #####  
#####  
#externalSort.threshold: 256MB  
  
#####  
##### Server timeout (10 min) #####  
#####  
spring.mvc.async.request-timeout: 600000  
  
#####  
##### DATABASE STATISTICS #####  
#####  
databaseStatistics : false  
  
#####  
##### CALLS GRAPH #####  
#####  
callGraph : false  
  
#####  
##### SQL SHIFT CODE POINT #####  
#####  
# Code point 384 match unicode character \u0180  
sqlCodePointShift : 384  
  
#####  
##### LOCK TIMEOUT RECORD #####  
#####  
# Blu4IV record lock timeout  
lockTimeout : 100  
  
#####  
##### REPORTS OUTPUT PATH #####  
#####  
reportOutputPath: reports  
  
#####
```



```

##### TASK EXECUTOR      ####
#####
taskExecutor:
  corePoolSize: 5
  maxPoolSize: 10
  queueCapacity: 50
  allowCoreThreadTimeOut: false

#####
##### PROGRAM NOT FOUND  ####
#####
stopExecutionWhenProgNotFound: false

#####
##### DISP DEFAULT VALUE (to be removed one day) ####
#####
defaultKeepExistingFiles: true

#####
##### JOBQUEUE CONFIGURATION  ####
#####
jobqueue:
  api.enabled: false
  impl: none # possible values: quartz, none
  schedulers: # list of schedulers
    -
      name: queue1
      threadCount: 5
    -
      name: queue2
      threadCount: 5

#####
##### QUERY BUILDING      ##
# useConcatCondition : false by default
# if true, in the query, the where condition is build with key concatenation ##
#####
# query.useConcatCondition: true
----

```

- Usa file a lunghezza variabile con i comandi LISTCAT

```

[**/*. *]
encoding=IBM930
reencoding=false

[global]
listcat.variablelengthpreprocessor.enabled=true
listcat.variablelengthpreprocessor.type=rdw
# use "rdw" if your .listcat file contains a set of records (RDW)
# use "bdw" if your .listcat file contains a set of blocks (bdw)

```

- Fornisci un valore dell'indicatore Null Byte nell'utilità LOAD/UNLOAD

```

# Unload properties
# For date/time: if use database configuration is enabled, formats are ignored
# For nbi; use hexadecimal syntax to specify the byte value
# - When the value is null in database : the value dumped to the file is filled by
  low value characters and the NBI is
# equal to the byte 6F (the ? character)
# - When the value is not null in database and the column is nullable: the NBI is
  equal to the byte 00 (low value) and NOT
# equal to the byte 40 (space)
unload:
  sqlCodePointShift: 0
  nbi:
    whenNull: "6F"
    whenNotNull: "00"
  useDatabaseConfiguration: false
  format:
    date: MM/dd/yyyy
    time: HH.mm.ss
    timestamp: yyyy-MM-dd-HH.mm.ss.SSSSSS

```

## Proprietà disponibili per l'applicazione principale

Questa tabella fornisce una visualizzazione esaustiva dei parametri chiave/valori.

Chiave	Type	Valore predefinito	Descrizione
<code>logging.config</code>	Path	<code>classpath: logback-main.xml</code>	Chiave standard per il riferimento al file di configurazione logback. Sono disponibili anche altre chiavi di registrazione standard.
<code>spring.jta.enabled</code>	booleano	<code>false</code>	Chiave standard. Se la modalità di supporto dell'origine dati non è <code>static-xa</code> , la configurazione automatica delle transazioni Spring JTA deve essere disabilitata.
<code>datasource.jicsDs + -driver-class-name + -url + -username + -password + -type</code>	Fonte di dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database Jics. In alternativa, l'uso dei segreti AWS è fortemente incoraggiato, come spiegato in <a href="#">xref:.. /configuration/configuration.adoc [Configurazione]</a> .
<code>datasource.bluesamDs + -driver-class-name + -url + -username + -password + -type</code>	Fonte dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database Blusam. In alternativa, l'uso dei segreti AWS è fortemente incoraggiato.

Chiave	Type	Valore predefinito	Descrizione
			ato, come spiegato in <a href="#">xref:.. /configuration/ configuration.adoc</a> [Configurazione].
<code>bluesam.disabled</code>	booleano	false	Se disabilitare completamente bluesam.
<code>bluesam.cache</code>	string		Se non è impostata, la cache bluesam non verrà utilizzata. I valori possibili (implementazioni della cache) sono ehcache e redis.
<code>forcedDate</code>	string		Forza la data alla data fornita, se presente.
<code>frozenDate</code>	booleano	true	Specifica se congelare la data. Si applica solo se <code>forcedDate</code> è impostata anche.
<code>externalSort.threshold</code>	dimensione dei dati (ad es. 12 MB)		La soglia di ordinamento: quando passare all'ordinamento esterno (unione).
<code>jics.parameters.datform</code>	string	MMDDYY	Il modulo della data.

Chiave	Type	Valore predefinito	Descrizione
<code>jics.initList`</code>	string		<p>L'elenco jics di inizializzazione, separato da virgole. Se presente, definisce i nomi degli elenchi separati da virgole da attivare all'avvio di tomcat tra gli elenchi CICS. Valore di esempio: <code>\$UUU,DFH\$IVPL,PEZ1</code></p> <p>Questo verrà trasferito a cascata ai gruppi contenuti in tali elenchi e alle relative definizioni di risorse sottostanti, che saranno quindi visibili al runtime. Vuoto per impostazione predefinita.</p>
<code>jics.parameters.applid</code>	string	VELOCITÀ	Applicato a per identificare l'applicazione in JICS (almeno 4 caratteri, nessuna lunghezza massima).
<code>jics.parameters.sysid</code>	string	CICS	L'identificazione del sistema (SYSID).
<code>jics.parameters.eibtrmid</code>	string	TERMINE	L'identificatore del terminale (massimo 4 caratteri, minimo 1).

Chiave	Type	Valore predefinito	Descrizione
<code>jics.parameters.userid</code>	string		L'id utente (massimo 8 caratteri, nessun minimo). Quando non viene fornito alcun valore (vuoto per impostazione predefinita), l'ID di sessione HTTP viene utilizzato come ID utente.
<code>jics.parameters.username</code>	string	IL MIO NOME UTENTE	Il nome utente (massimo 10 caratteri, minimo 1).
<code>jics.parameters.netname</code>	string	IL MIO NOME DI RETE	Il nome della rete (massimo 8 caratteri, 1 minimo).
<code>jics.parameters.opid</code>	string	XXX	L'identificazione dell'operatore a 3 caratteri.
<code>jics.parameters.jobname`</code>	string	NOME DEL LAVORO	Il nome del lavoro.
<code>jics.parameters.sysname</code>	string	SYSNAME	Il nome del sistema AS400 (sysname).
<code>jics.parameters.cwa.length</code>	number	0	La lunghezza dell'area di lavoro comune (cwa).

Chiave	Type	Valore predefinito	Descrizione
<code>jics.parameters.charset</code>	string	CP-037	Set di caratteri JICS utilizzato a livello globale.
<code>jics.parameters.tsqimpl</code>	string	bluesam	Implementazione JICS Temporary Storage Queue (TSQ) (i valori consentiti sono//) bluesam memory redis
<code>jics.queues.ts.redis.hostname</code>	string	1270.0.1	Il nome host del server redis jics cache.
<code>jics.queues.ts.redis.port</code>	number	6379	La porta del server redis jics cache.
<code>jics.queues.ts.redis.password</code>	string	redis	La password per il server redis jics cache.
<code>jics.queues.ts.redis.username</code>	string		Il nome utente per il server jics cache redis. L'impostazione predefinita è vuota (nessun nome utente).
<code>jics.queues.ts.redis.mode</code>	string	indipendente, autonomo	La modalità cache jics. I valori possibili sono standalone o cluster. L'impostazione predefinita è standalone .

Chiave	Type	Valore predefinito	Descrizione
lockTimeout	number	500	Il timeout di blocco, in millisecondi.
sqlCodePointShift	number		Facoltativo. Lo spostamento dei punti del codice sql. Sposta il codepoint per i caratteri di controllo che potremmo incontrare durante la migrazione dei dati rdbms legacy a un rdbms moderno. Ad esempio, è possibile specificare che corrisponda al carattere unicode. 384 \u0180
sqlIntegerOverflowAllowed	booleano	false	Specifica se consentire l'overflow di numeri interi SQL, ovvero se è consentito inserire valori più grandi nella variabile host.



Chiave	Type	Valore predefinito	Descrizione
database.cursor.overflow.allowed	booleano	true	Specifica se consentire l'overflow del cursore. Imposta per true eseguire una chiamata successiva sul cursore indipendentemente dalla sua posizione. Imposta false per verificare se il cursore si trova nell'ultima posizione prima di eseguire una chiamata successiva sul cursore. Abilita solo se il cursore è SCORREVOLE (SENSITIVE o INSENSITIVE).
reportOutputPath	string	/reports	Il percorso di output del report.
spring.session.store-type	string	nessuno	La cache della sessione per ambienti ad alta disponibilità. I valori possibili sono none o redis. L'impostazione predefinita è none.

Chiave	Type	Valore predefinito	Descrizione
<code>stopExecutionWhenProgramNotFound</code>	booleano	true	Specifica se interrompere l'esecuzione se non viene trovato un programma. Se impostato su <code>true</code> , interrompe l'esecuzione se non viene trovato un programma.
<code>forceHR</code>	booleano	false	Specifica se utilizzare Human Readable SYSPRINT, sulla console o sull'output del file.
<code>rollbackOnRTE</code>	booleano	false	Specifica se eseguire il rollback della transazione implicita dell'unità di esecuzione e sulle eccezioni di runtime.
<code>sctThreadLimit</code>	Long	5	Il limite dei thread per l'attivazione degli script.

Chiave	Type	Valore predefinito	Descrizione
<code>dataSimplifier.onInvalidNumericData</code>	string	rifiutare	Come reagire quando si decodificano dati numerici non validi. I valori consentiti sono <code>reject</code> /// <code>tolerates paces</code> . <code>tolerates paceslowvalues</code> <code>toleratemo</code> st L'impostazione predefinita è <code>reject</code> .
<code>filesDirectory</code>	string		La directory per i file di input/output in batch.
<code>ims.messages.extendedSize</code>	booleano	false	Speciifica se impostare <code>ExtendedSize</code> sui messaggi <code>ims</code> .
<code>defaultKeepExistingFiles</code>	booleano	false	Speciifica se impostare il valore precedente predefinito del set di dati.
<code>jics.db.ddlScriptLocation</code>	string		La posizione dello script <code>ddl</code> <code>Jics</code> . Consente di avviare lo schema del database <code>jics</code> utilizzando uno script <code>sql</code> . Vuoto per impostazione predefinita. Ad esempio, <code>./jics/sql/jics.sql</code> .

Chiave	Type	Valore predefinito	Descrizione
<code>jics.db.schemaTestQueryLocation</code>	string		Posizione del file sql che dovrebbe contenere una query univoca che restituisce il numero di oggetti nello schema jics (se presente).
<code>jics.db.dataScriptLocation</code>	string		Posizione dello script <code>initJics.sql</code> , preparato da Analyzer in base all'analisi delle esportazioni CSD dal mainframe.
<code>jics.db.dataTestQueryLocation</code>	string		Posizione di uno script sql contenente una singola query sql che dovrebbe restituire un numero di oggetti (ad esempio: conteggio del numero di record nella tabella del programma jics). Se il conteggio è uguale a 0, il database verrà caricato utilizzando <code>jics.db.dataScriptLocation</code> lo script, altrimenti il caricamento del database verrà saltato.

Chiave	Type	Valore predefinito	Descrizione
<code>jics.data.dataJsonInitLocation</code>	string		
<code>jics.xa.agent.timeout</code>	number		
<code>query.useConcatCondition</code>	booleano	false	Specifica se la condizione della chiave è creata o meno mediante la concatenazione di chiavi.
<code>system.qdecfmt</code>	string		
<code>disposition.checkexistence</code>	booleano	false	Specifica se rilasciar e un controllo sull'esistenza dei file per Dataset with DISP SHR o OLD.
<code>useControlMVariable</code>	booleano	false	Specifica se utilizzare e la specifica Control-M per la sostituzione delle variabili.
<code>card.encoding</code>	string	CP1145	Codifica delle carte: da utilizzare e con <code>useControlMVariable`</code> .

Chiave	Type	Valore predefinito	Descrizione
<code>mapTransfo.prefixes</code>	string	<code>&amp;,@,%%</code>	Elenco di prefissi da utilizzare per la trasformazione delle variabili ControlM. Ciascuno separato da una virgola.
<code>checkinputfilesize</code>	booleano	false	Specifica se rilasciar e un segno di spunta se la dimensione del file è un multiplo della dimensione del record.
<code>stepFailWhenAbend</code>	booleano	true	Specifica se alzare un abend se un passaggio ha esito negativo o completa l'esecuzione.
<code>bluesam.fileLoading.commitInterval</code>	number	100000	L'intervallo di commit di bluesam.
<code>uppercaseUserInput</code>	booleano	true	Specifica se l'input dell'utente deve essere in maiuscolo.
<code>jhdb.lterm</code>	string		Consentono di forzare un ID di terminale logico comune nel caso di un'emulazione IMS. Se non è impostato, viene utilizzato SessionID.

Chiave	Type	Valore predefinito	Descrizione
jhdb.identificatio nCardData	string		Utilizzato per codificar e alcuni «dati della carta di identificazione dell'operatore» nel campo MID indicato dal parametro CARD. Vuoto per impostazi one predefinita, nessuna restrizione di input.
encoding	string	ASCII	La codifica utilizzata nei progetti (non nei file groovy). Prevede una codifica CP1047 valida,,,... IBM930 ASCII UTF-8
cl.config uration.c ontext.en coding	string	CP297	La codifica dei file CL. Prevede una codifica validaCP1047,,IBM930... ASCII UTF-8 Il valore predefinito è CP297
cl.zonedMode	string	EBCDIC_STRICT	La modalità per la codifica o la decodific a dei comandi del linguaggio di controllo (CL). I valori consentit i sonoEBCDIC_ST RICT //EBCDIC_MO DIFIED .AS400

Chiave	Type	Valore predefinito	Descrizione
<code>ims.programs</code>	string		Elenco dei programmi IMS da utilizzare. Separare ogni parametro con un punto e virgola (;) e ogni transazione con una virgola (,). Ad esempio:PCP008,PC T008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;
<code>jhdb.configuration.context.encoding</code>	string	CP297	La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida,,,,... CP1047 IBM930 ASCII UTF-8
<code>jhdb.metadata.extrapath</code>	string	file:./configurazione/	Un parametro di configurazione che specifica una cartella principale aggiuntiva, specifica per il runtime, per le cartelle psbs e dbds.



Chiave	Type	Valore predefinito	Descrizione
jhdb.checkpointPersistence	string	nessuno	<p>La modalità di persistenza del checkpoint. I valori consentiti sono none//add. end Si usa add per mantenere i checkpoint quando ne viene creato uno nuovo e aggiunto al registro. Usa per end persistere il checkpoint alla chiusura del server. Qualsiasi altro valore disabilita la persistenza. Tieni presente che ogni volta che viene aggiunto un nuovo checkpoint al registro, tutti i checkpoint esistenti verranno serializzati e il file verrà cancellato. Non è un'aggiunta ai dati esistenti nel file. Quindi, a seconda del numero di checkpoint, può avere alcuni effetti sulle prestazioni.</p>

Chiave	Type	Valore predefinito	Descrizione
jhdb.chec kpointPath	string	file:. /configurazione/	In caso contrario jhdb.chec kpointPer sistence , questo parametro consente di impostare il percorso di persistenza del checkpoint (posizione e di archiviazione del file checkpoint t.dat), tutti i dati dei checkpoint contenuti nel registro vengono serializzati e sottoposti a backup in un file (checkpoint.dat) situato nella cartella fornita. none Nota che solo i dati del checkpoint (ScriptID , StepID, posizione del database e area del checkpoint) sono interessati da questo backup.
jhdb.navi gation.ca chenexts	number	5000	La durata della cache (in millisecondi) utilizzata nella navigazione gerarchica per un RDBMS.

Chiave	Type	Valore predefinito	Descrizione
<code>jhdb.use-db-prefix</code>	booleano	true	Specifica se abilitare un prefisso di database nella navigazione gerarchica per un RDBMS.
<code>jhdb.query.limitJoinUsage</code>	booleano	true	Specifica se utilizzare il parametro limit join usage sui grafici RDBMS.
<code>taskExecutor.corePoolSize</code>	number	5	Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per configurare la dimensione del pool principale.
<code>taskExecutor.maxPoolSize</code>	number	10	Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per impostare la dimensione massima del pool (numero massimo di thread paralleli).

Chiave	Type	Valore predefinito	Descrizione
<code>taskExecutor.queueCapacity</code>	number	50	Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per impostare la dimensione della coda. (= numero massimo di transazioni in sospeso quando viene raggiunto) <code>taskExecutor.maxPoolSize</code>
<code>taskExecutor.allowCoreThreadTimeOut</code>	booleano	false	Specifica se consentire il timeout dei thread principali in JCIS. Ciò consente l'aumento e la riduzione dinamica anche in combinazione con una coda diversa da zero (poiché la dimensione massima del pool aumenterà solo quando la coda sarà piena).
<code>jics.runUnitLauncherPool.enable</code>	booleano	false	Specifica se attivare il run unit launcher pool in JICS.

Chiave	Type	Valore predefinito	Descrizione
<code>jics.runUnitLauncherPool.size</code>	number	20	La dimensione del pool di avvio dell'unità di esecuzione in JICS.
<code>jics.runUnitLauncherPool.validationInterval</code>	number	1000	L'intervallo di convalida del pool di avvio dell'unità di esecuzione in JICS, espresso in millisecondi.
<code>spring.aws.application.credentials</code>	string	null	Carica le credenziali AWS dal file dei profili delle credenziali in JICS.
<code>jics.queues.sqs.region</code>	string	eu-west-1	La regione AWS per AWS Simple Queue Service, utilizzata in JICS.
<code>mq.queues.sqs.region</code>	string	eu-west-3	La regione AWS per il servizio AWS SQS MQ.
<code>quartz.scheduler.standby-if-error</code>	booleano	false	Specifica se attivare l'esecuzione del lavoro se lo scheduler del lavoro è in modalità standby. Se vero, quando è abilitata l'esecuzione del lavoro non viene attivata.

Chiave	Type	Valore predefinito	Descrizione
databaseStatistics	booleano	false	Specifica se consentire ai costruttori di SQL di raccogliere e visualizzare informazioni statistiche.
dbDateFormat	string	yyyy-MM-dd	Il formato della data di destinazione del db.
dbTimeFormat	string	HH: mm: ss	Il formato dell'ora di destinazione del db.
dbTimestampFormat	string	aaaa-mm-gg HH:mm:ss.ssssss	Il formato del timestamp di destinazione del db.
dateTimeFormat	string	ISO	dateTimeFormat Descrive come trasferire il tipo di data, ora e ora del database in entità semplificatrici di dati. I valori consentiti sono///ISOEUREURUSALOCAL
localDateFormat	string		Elenco di formati di data locali. Separa ogni formato con. \
localTimeFormat	string		Elenco dei formati dell'ora locale. Separa ogni formato con \

Chiave	Type	Valore predefinito	Descrizione
localTime stampFormat	string		Elenco dei formati di timestamp locali. Separa ogni formato con. \
pgmDateFormat	string	yyyy-MM-dd	Il formato della data e dell'ora.
pgmTimeFormat	string	HH.mm.ss	Il formato dell'ora usato per l'esecuzione di pgm (programmi).
pgmTimest ampFormat	string	yyyy-MM-GD- hh.mm.ss.ssssss	Il formato del timestamp.
cacheMetadata	booleano	true	Specifica se memorizzare nella cache i metadati del database.
forceDisa bleSQLTri mStringType	booleano	false	Specifica se disabilitare il taglio di tutti i parametri della stringa sql.
fetchSize	number		Il valore fetchSize per i cursori. Da utilizzare quando si recuperano dati utilizzando blocchi tramite utilità load/unload.
check-groovy- file	booleano	true	Specifica se controllare il contenuto dei file Groovy prima della registrazione.

Chiave	Type	Valore predefinito	Descrizione
<code>qtemp.uuid.length</code>	number	9	La lunghezza dell'ID univoco di QTEMP.
<code>qtemp.dblog</code>	booleano	false	Se abilitare la registrazione del database QTEMP.
<code>qtemp.cleanup.threshold.hours</code>	number	0	Per specificare quando è abilitato <code>qtemp.dblog</code> . La durata della partizione db (in ore).
<code>sort.function</code>	string		Il nome della funzione di ordinamento per il database blu4iv.

## Proprietà disponibili per applicazioni web opzionali

A seconda dell'applicazione modernizzata in uso, potrebbe essere necessario configurare una o più applicazioni Web opzionali che supportino dipendenze come z/OS, AS/400 o IMS/MFS. Le tabelle seguenti contengono elenchi dei parametri chiave/valore disponibili per la configurazione di ogni applicazione Web opzionale.

`gapwalk-utility-pgm.guerra`

Questa applicazione web opzionale contiene il supporto per i programmi di utilità Z/OS.

Questa tabella fornisce una visualizzazione esaustiva dei parametri chiave/valori per questa applicazione.

Chiave	Type	Valore predefinito	Descrizione
<code>logging.config</code>	Path	classpath: logback-utility.xml	Chiave standard per il riferimento al file di configura



Chiave	Type	Valore predefinito	Descrizione
			zione logback. Sono disponibili anche altre chiavi di registrazione standard.
<code>spring.jta.enabled</code>	booleano	false	Chiave standard. Se la modalità di supporto dell'origine dati non è static-xa, la configurazione automatica delle transazioni Spring JTA deve essere disabilitata.
<code>spring.datasource.primary.jndi-name</code>	string	jdbc/primario	Il nome jndi (Java Naming And Directory Interface) per l'origine dati primaria, se si utilizza JNDI.

Chiave	Type	Valore predefinito	Descrizione
<pre>primary.datasource + -driver-class-name + -url + -username + - password</pre>	Fonte dati Spring standard con sottochiavi		<p>Contiene le informazioni di connessione per il database dell'applicazione, se non si utilizza JNDI. Deve avere la stessa configurazione del file yml dell'applicazione modernizzata.</p> <p>In alternativa, l'uso dei segreti AWS è fortemente incoraggiato, come spiegato in <a href="#">xref:.. /configuration/ configuration.adoc [Configurazione]</a>.</p>
encoding	string	ASCII	La codifica utilizzata nei programmi di utilità. Prevede una codifica valida CP1047,,IBM930... ASCII UTF-8
sysPunchEncoding	string	ASCII	Il set di caratteri di codifica syspunch. Prevede una codifica valida,,... CP1047 IBM930 ASCII UTF-8

Chiave	Type	Valore predefinito	Descrizione
zonedMode	string	EBCDIC_STRICT	La modalità per codificare o decodificare i tipi di dati suddivisi in zone. I valori consentiti sono//EBCDIC_STRICT . EBCDIC_MODIFIED AS400
unload.chunkSize	number	0	Dimensione del blocco utilizzata per l'utilità di scarico.
unload.sqlCodePointShift	number	0	Il codice SQL pointshift per l'utilità di scarico. Esegue il processo di cambio dei caratteri. Richiesto quando il database di destinazione di DB2 è Postgresql.
unload.columnFiller	string	spazio	Il riempitivo di colonne dell'utilità unload.
unload.variableCharIsNull	booleano	false	Usa questo parametro nel programma INFTILB, se impostato su true allora tutti i campi non annullabili con valori vuoti (spazio) restituiscono una stringa vuota.

Chiave	Type	Valore predefinito	Descrizione
<code>unload.us eDatabase Configuration</code>	booleano	false	Specifica se utilizzare la configurazione di data o ora da applicazioni-main.yml nell'utilità unload.
<code>unload.fo rmat.date</code>	string	MM/dd/yyyy	Se abilitato, unload.us eDatabase Configuration il formato della data da utilizzare nell'utilità di scarico.
<code>unload.fo rmat.time</code>	string	H.mm.ss	Se unload.us eDatabase Configuration abilitato, il formato dell'ora da utilizzare nell'utilità di scaricamento.
<code>unload.fo rmat.timestamp</code>	string	YYYY-MM-GD- hh.mm.ss.ssssss	Se abilitato, il formato del timestamp da utilizzare nell'utilità di scaricamento. unload.us eDatabase Configuration
<code>unload.nb i.whenNull</code>	esadecimale	6F	Il valore dell'indicatore Null Byte (nbi) da aggiungere quando il valore del database è nullo.

Chiave	Type	Valore predefinito	Descrizione
<code>unload.nbi.whenNotNull</code>	esadecimale	00	Il valore Null Byte Indicator (nbi) da aggiungere quando il valore del database non è nullo.
<code>unload.nbi.writeNullIndicator</code>	booleano	false	Specifica se scrivere l'indicatore null nel file di output di scaricamento.
<code>unload.fetchSize</code>	number	0	Consente di regolare la dimensione del recupero quando si gestiscono i cursori nell'utilità di scarico.
<code>treatLargeNumberAsInteger</code>	booleano	false	Specifica se trattare i numeri grandi come Integer. Per impostazione predefinita, vengono BigDecimal trattati come tali.
<code>load.batchSize</code>	number	0	La dimensione del batch dell'utilità di caricamento.
<code>load.format.localDate</code>	string	gg.mm.aaaa\ gg/MM/aaaa\ aaaa-MM-gg	Il formato di data locale dell'utilità di caricamento da utilizzare.

Chiave	Type	Valore predefinito	Descrizione
<code>load.form at.localTime</code>	string	HH:mm:ss\ hh.mm.ss	Il formato dell'ora locale dell'utilità di caricamento da utilizzare.
<code>load.form at.dbDate</code>	string	yyyy-MM-dd	Il formato del database dell'utilità di caricamento da utilizzare.
<code>load.form at.dbTime</code>	string	HH:mm:ss	Il tempo di utilizzo del database dell'utilità di caricamento.
<code>load.sqlC odePointShift</code>	number	0 s	Il codice SQL pointshift per l'utilità di caricamento. Eseguire il processo di cambio dei caratteri. Richiesto quando il database di destinazione di DB2 è Postgresql.
<code>forcedDate</code>	string		Impone la data alla data fornita, se presente.
<code>frozenDate</code>	booleano	true	Specifica se congelare la data. Si applica solo se <code>forcedDate</code> è impostata anche.

Chiave	Type	Valore predefinito	Descrizione
<code>jcl.type</code>	string	mvs	tipo di file.jcl. I valori consentiti sono/j c l. vse I comandi PRINT/REPRO dell'utilità IDCAMS restituiscono 4 se il file è vuoto per jcl non vse.
<code>hasGraphic</code>	booleano	false	Se l'utilità INFUTILB deve gestire le colonne GRAPHIC DB2.

### gapwalk-cl-command.guerra

Questa applicazione web opzionale contiene il supporto per i programmi di utilità AS/400.

Questa tabella fornisce una panoramica esaustiva dei parametri chiave/valori per questa applicazione.

Chiave	Type	Valore predefinito	Descrizione
<code>logging.config</code>	Path	classpath: logback-utility.xml	Chiave standard per il riferimento al file di configurazione logback. Sono disponibili anche altre chiavi di registrazione standard.
<code>spring.jta.enabled</code>	booleano	false	Chiave standard. Se la modalità di supporto dell'origine dati non è static-xa

Chiave	Type	Valore predefinito	Descrizione
			, la configurazione automatica delle transazioni Spring JTA deve essere disabilitata.
<code>spring.datasource.primary.jndi-name</code>	string	jdbc/primario	Il nome jndi (Java Naming And Directory Interface) per l'origine dati primaria, se si utilizza JNDI.
<code>primary.datasource + -driver-class-name + -url + -username + -password</code>	Fonte dati Spring standard con sottochiavi		<p>Contiene le informazioni di connessione per il database dell'applicazione, se non si utilizza JNDI. Deve avere la stessa configurazione del file yml dell'applicazione modernizzata.</p> <p>In alternativa, l'uso dei segreti AWS è fortemente incoraggiato, come spiegato in <a href="#">xref:.. /configuration/ configuration.adoc [Configurazione]</a>.</p>
<code>encoding</code>	string	ASCII	La codifica utilizzata nei programmi di utilità. Prevede una codifica valida CP1047,,IBM930... ASCII UTF-8



Chiave	Type	Valore predefinito	Descrizione
zonedMode	string	EBCDIC_STRICT	La modalità per codificare o decodificare i tipi di dati suddivisi in zone. I valori consentiti sono//EBCDIC_STRICT . EBCDIC_MODIFIED AS400
commands-off	string		Elenco di comandi da disattivare, separati da virgole. I valori consentiti sono PGM_BASIC RCVMSGSNDRCVF,CHGVAR,,QCF,DLTOVRCPYF,SNDDTAQ. Utile quando si desidera disabilitare o sovrascrivere un programma esistente . PGM_BASIC è un programma di velocità specifico progettato per scopi di debug.

gapwalk-hierarchical-support.guerra

Questa applicazione web opzionale contiene il supporto per le transazioni IMS/MFS.

Questa tabella fornisce una visualizzazione esaustiva dei parametri chiave/valori per questa applicazione.

Chiave	Type	Valore predefinito	Descrizione
<code>logging.config</code>	Path	<code>classpath: logback-utility.xml</code>	Chiave standard per il riferimento al file di configurazione logback. Sono disponibili anche altre chiavi di registrazione standard.
<code>spring.jta.enabled</code>	booleano	<code>false</code>	Chiave standard. Se la modalità di supporto dell'origine dati non è <code>static-xa</code> , la configurazione automatica delle transazioni Spring JTA deve essere disabilitata.
<code>jhdb.configuration.context.encoding</code>	string		La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida,,,... CP1047 IBM930 ASCII UTF-8
<code>jhdb.checkpointPersistence</code>	string	nessuno	La modalità di persistenza del checkpoint. I valori consentiti sono <code>none//add.end</code> . Si usa <code>add</code> per mantenere i checkpoint quando ne viene creato uno

Chiave	Type	Valore predefinito	Descrizione
			nuovo e aggiunto al registro. Usa per end persistere il checkpoint alla chiusura del server. Qualsiasi altro valore disabilita la persistenza. Tieni presente che ogni volta che viene aggiunto un nuovo checkpoint al registro, tutti i checkpoint esistenti verranno serializzati e il file verrà cancellato. Non è un'aggiunta ai dati esistenti nel file. Quindi, a seconda del numero di checkpoint, può avere alcuni effetti sulle prestazioni.

## Configurare l'autenticazione per le applicazioni Gapwalk

Questa sezione descrive come configurare l'autenticazione OAuth2 per le applicazioni Gapwalk utilizzando un Identity Provider (IdP) come Cognito, Azure AD, Keycloak, ecc.

### Argomenti

- [Prerequisito](#)
- [Configurazione di Amazon Cognito](#)
- [Integrazione di Amazon Cognito nell'applicazione Gapwalk](#)

## Prerequisito

In questo tutorial utilizzeremo Amazon Cognito come IdP e PlanetDemo come progetto modernizzato.

Puoi utilizzare qualsiasi altro provider di identità esterno. Le ClientRegistration informazioni devono essere ottenute dal tuo IdP e sono necessarie per l'autenticazione gapwalk. Per ulteriori informazioni, consulta la documentazione ufficiale del tuo IdP.

Le ClientRegistration informazioni:

client-id

l'id di ClientRegistration, nel nostro esempio sarà PlanetDemo.

segreto del cliente

il segreto del tuo cliente.

endpoint di autorizzazione

L'URI dell'endpoint di autorizzazione per il server di autorizzazione.

endpoint del token

L'URI Token Endpoint per il server di autorizzazione.

endpoint jwks

L'URI utilizzato per ottenere la chiave Web JSON (JWK) contenente le chiavi per la convalida della firma Web JSON emessa dal server di autorizzazione.

URI di reindirizzamento

URI a cui il server di autorizzazione reindirizza l'utente finale se l'accesso viene concesso.

## Configurazione di Amazon Cognito

Per prima cosa creeremo e configureremo un pool di utenti e utenti Amazon Cognito che utilizzeremo con la nostra applicazione gapwalk distribuita a scopo di test.

### Note

Se utilizzi un altro IdP puoi saltare questo passaggio.

## Crea un pool di utenti

1. Accedi ad Amazon Cognito in AWS Management Console ed esegui l'autenticazione utilizzando le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli Create a User Pool (Crea un bacino d'utenza).
4. In Configura l'esperienza di accesso, mantieni il tipo di provider predefinito del pool di utenti Cognito. Puoi scegliere una o più opzioni di accesso al pool di utenti di Cognito; per ora, scegli Nome utente, quindi scegli Avanti.
5. In Configura i requisiti di sicurezza, mantieni le impostazioni predefinite e disabilita l'autenticazione a più fattori scegliendo Nessuna MFA, quindi scegli Avanti.
6. Disattiva Abilita l'autoregistrazione come misura di sicurezza e scegli Avanti.
7. Scegli Invia email con Cognito. Seleziona Avanti.
8. In Integra la tua app, scegli un nome per il tuo pool di utenti. Nelle pagine di autenticazione ospitate, scegli Usa l'interfaccia utente ospitata da Cognito.
9. Per semplicità, in Dominio, scegli Usa un dominio Cognito e inserisci un prefisso di dominio, ad esempio. `https://planetsdemo` L'app demo deve essere aggiunta come client.
  1. Nel client iniziale dell'app, scegli Client riservato. Inserisci il nome del client dell'app, ad esempio ``planetsdemo`` e scegli Genera un segreto client.
  2. In URL di callback consentito, inserisci l'URL a cui reindirizzare l'utente dopo l'autenticazione. È inoltre possibile utilizzare un URL temporaneo `http://localhost:8080/planetsdemo`, che può essere modificato in un secondo momento.
  3. Mantieni i valori predefiniti nelle sezioni Impostazioni avanzate del client dell'app e Attribuisce autorizzazioni di lettura e scrittura.
  4. Seleziona Avanti.
10. In Rivedi e crea, verifica le tue scelte, quindi scegli Crea pool di utenti.

Per ulteriori informazioni, consulta [Creare un pool di utenti](#).

## Creazione di utenti

Poiché l'autoregistrazione è disabilitata, crea un utente Amazon Cognito. Accedi ad Amazon Cognito in AWS Management Console Scegli il pool di utenti che hai creato, quindi in Utenti scegli Crea utente.

In Informazioni utente, scegli Invia un invito via e-mail, inserisci un nome utente e un indirizzo e-mail e scegli Genera una password. Selezionare Create user (Crea utente).

## Integrazione di Amazon Cognito nell'applicazione Gapwalk

Ora che il pool di utenti e gli utenti di Amazon Cognito sono pronti, vai al `main-application.yml` file della tua applicazione modernizzata e aggiungi il codice seguente:

```
spring:
  security:
    oauth2:
      client:
        registration:
          cognito:
            client-id: client-id
            client-name: client-name
            client-secret: client-secret
            provider: cognito
            authorization-grant-type: authorization_code
            scope: openid
            redirect-uri: "{baseUrl}/login/oauth2/code/{registrationId}"
        provider:
          cognito:
            issuerUri: ${gapwalk-application.security.issuerUri}
            authorization-uri: ${gapwalk-application.security.domainName}/oauth2/
authorize
  jwks.json
    token-uri: ${gapwalk-application.security.domainName}/oauth2/token
    user-name-attribute: cognito:username
  resourceserver:
    jwt:
      jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/jwks.json

gapwalk-application.security: enabled
gapwalk-application.security.identity: oauth

gapwalk-application.security.issuerUri: https://cognito-idp.region.amazonaws.com/pool-id
gapwalk-application.security.domainName: your-cognito-domain
```

Sostituisci i seguenti segnaposto come descritto:

1. Accedi ad Amazon Cognito in AWS Management Console ed esegui l'autenticazione utilizzando le tue credenziali. AWS
2. Scegli User Pools e scegli il pool di utenti che hai creato. Puoi trovare il tuo *pool-id* in User pool ID.
3. Scegli l'integrazione delle app dove puoi trovare la tua app *your-cognito-domaine* vai su App client e analisi e scegli la tua app.
4. In App client: YourApp puoi trovare il **nome del client, l'id del cliente e il segreto del cliente (Mostra il segreto del cliente)**.
5. *region-id corrisponde all'ID* della regione in cui hai creato l'utente e il pool di utenti di Amazon Cognito eu-west-3, ad esempio.
6. Per redirect-uri, inserisci l'URI a cui reindirizzare l'utente dopo l'autenticazione.

Puoi implementare subito la tua applicazione Gapwalk e utilizzare l'utente creato in precedenza per accedere alla tua app.

#### Note

Se durante l'accesso ricevi un errore con l'eccezione «Attributo 'cognito:username' mancante negli attributi», rimuovi la seguente riga:: cognito:username user-name-attribute

## AWS API Blu Age Runtime

AWS Blu Age Runtime utilizza diverse applicazioni Web per esporre gli endpoint REST, fornendo modi per interagire con le applicazioni modernizzate utilizzando i client REST (ad esempio chiamando i lavori utilizzando uno scheduler).

Lo scopo di questo documento è elencare gli endpoint REST disponibili, fornendo dettagli su:

- Il loro ruolo
- Il modo di usarli correttamente

L'elenco degli endpoint è organizzato in categorie, a seconda della natura del servizio fornito e dell'applicazione web che espone gli endpoint.

[Partiamo dal presupposto che tu abbia già una conoscenza di base sull'uso degli endpoint REST \(utilizzando strumenti dedicati come POSTMAN, Thunder Client, CURL, browser web, ecc...\) o scrivendo il tuo codice per effettuare una chiamata API.](#)

## Argomenti

- [Creazione di URL](#)
- [Applicazione Gapwalk](#)
- [Endpoint REST della Blusam Application Console](#)
- [Console applicativa JICS](#)
- [Strutture dati](#)

## Creazione di URL

Ogni applicazione web riportata di seguito definisce un percorso root, condiviso da tutti gli endpoint. Ogni endpoint aggiunge quindi il proprio percorso dedicato. L'URL risultante da utilizzare è il risultato della concatenazione dei percorsi. Ad esempio, considerando il primo endpoint dell'applicazione Gapwalk, abbiamo:

- `/gapwalk-application` per il percorso principale dell'applicazione web.
- `/scripts` per il percorso dedicato dell'endpoint.

L'URL risultante da utilizzare sarà `http://server:port/gapwalk-application/scripts`

`server`

punta al nome del server (quello che ospita l'applicazione web specificata).

`port`

la porta esposta dal server.

## Applicazione Gapwalk

Gli endpoint per l'applicazione web Gapwalk utilizzano il percorso root. `/gapwalk-application`

## Argomenti

- [Endpoint correlati ai job Batch \(JCL modernizzati e simili\)](#)



- [Metriche, endpoint](#)
- [Altri endpoint](#)
- [Endpoint relativi a Job Queues](#)

## Endpoint correlati ai job Batch (JCL modernizzati e simili)

I processi Batch possono essere eseguiti in modo sincrono o asincrono (vedere i dettagli di seguito). I processi Batch vengono eseguiti utilizzando script groovy che sono il risultato della modernizzazione degli script legacy (JCL).

### Argomenti

- [Elenca gli script distribuiti](#)
- [Avvia uno script in modo sincrono](#)
- [Avvia uno script in modo asincrono](#)
- [Elenco degli script attivati](#)
- [Recupero dei dettagli sull'esecuzione del lavoro](#)
- [Elenco degli script avviati in modo asincrono che possono essere eliminati](#)
- [Elenco degli script avviati in modo sincrono che possono essere eliminati](#)
- [Uccidere una determinata esecuzione di lavoro](#)
- [Elenco dei checkpoint esistenti per la riavviabilità](#)
- [Riavvio di un lavoro \(in modo sincrono\)](#)
- [Riavvio di un lavoro \(in modo asincrono\)](#)
- [Impostazione del limite di thread per le esecuzioni di lavori asincroni](#)

### Elenca gli script distribuiti

- Metodo supportato: GET
- Percorso: /scripts
- Argomenti: nessuno
- Questo endpoint restituisce l'elenco degli script groovy distribuiti sul server, come stringa. Questo endpoint è destinato principalmente all'uso da un browser Web, poiché la String risultante è una pagina HTML, con collegamenti attivi (un collegamento per script avviabile - vedi esempio di seguito).

## Risposta di esempio:

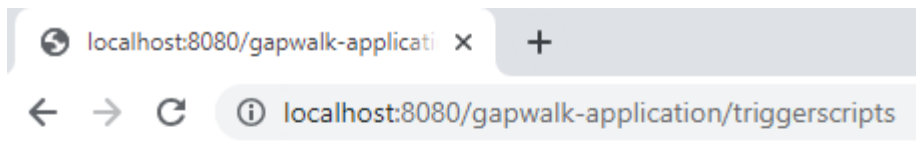
```
<p><a href=./script/COMBTRAN>COMBTRAN</a></p><p><a href=./script/CREASTMT>CREASTMT</a></p><p><a href=./script/INTCALC>INTCALC</a></p><p><a href=./script/POSTTRAN>POSTTRAN</a></p><p><a href=./script/REPROC>REPROC</a></p><p><a href=./script/TRANBKP>TRANBKP</a></p><p><a href=./script/TRANREPT>TRANREPT</a></p><p><a href=./script/functions>functions</a></p>
```

### Note

I collegamenti rappresentano l'URL da utilizzare per avviare ogni script elencato in modo sincrono.

- Metodo supportato: GET
- Percorso: /triggerscripts
- Argomenti: nessuno
- Questo endpoint restituisce l'elenco degli script groovy distribuiti sul server, come stringa. Questo endpoint è destinato principalmente all'uso da un browser Web, poiché la String risultante è una pagina HTML, con collegamenti attivi (un collegamento per script avviabile - vedi esempio di seguito).

A differenza della precedente risposta dell'endpoint, i link rappresentano l'URL da utilizzare per avviare ogni script elencato in modo asincrono.



[COMBTRAN](#)

[CREASTMT](#)

[INTCALC](#)

[POSTTRAN](#)

[REPROC](#)

[TRANBKP](#)

[TRANREPT](#)

## Avvia uno script in modo sincrono

Questo endpoint ha due varianti con percorsi dedicati per l'utilizzo di GET e POST (vedi sotto).

- Metodo supportato: GET
- Percorso: `/script/{scriptId:.+}`
- Metodo supportato: POST
- Percorso: `/post/script/{scriptId:.+}`
- Argomenti:
  - identificatore dello script da avviare
  - opzionalmente: parametri da passare allo script, utilizzando i parametri di richiesta (visti come `aMap<String,String>`). I parametri forniti verranno aggiunti automaticamente ai [collegamenti dello script groovy](#) invocato.
- La chiamata avvierà lo script con l'identificatore specificato, utilizzando parametri aggiuntivi se forniti e attenderà il completamento dell'esecuzione dello script prima di restituire un messaggio () `String` che sarà:
  - «Fatto». (se l'esecuzione del lavoro è avvenuta senza intoppi).
  - Un messaggio di errore JSON con dettagli su cosa è andato storto durante l'esecuzione del lavoro. Ulteriori dettagli possono essere recuperati dai log del server, per capire cosa è andato storto nell'esecuzione del lavoro.

```
{
  "exitCode": -1,
  "stepName": "STEP15",
  "program": "CBACT04C",
  "status": "Error"
}
```

Guardando i log del server, possiamo capire che si tratta di un problema di distribuzione (il programma previsto non è stato distribuito correttamente, quindi non può essere trovato, facendo fallire l'esecuzione del lavoro):

```

2023-06-09 10:27:28 default INFO - c.n.b.g.r.s.BatchWebController - --> executing script INTCALC
2023-06-09 10:27:28 default INFO - c.n.b.g.r.s.BatchWebController - Bound jobContext 419695287 - GDGEventsQueueHandler :907380469
2023-06-09 10:27:28 default INFO - c.n.b.g.r.s.ScriptControlTower - Added jobExecutor [a65c2791-864f-43c9-972a-b5f2353389e6] to Sync Script Control Tower.
2023-06-09 10:27:28 default INFO - c.n.b.g.r.s.j.sJobExecutor - a65c2791-864f-43c9-972a-b5f2353389e6 - worker :Thread-26 [1547512424]
2023-06-09 10:27:28 default INFO - c.n.b.g.r.s.j.sJobExecutor - Triggered script: INTCALC - [a65c2791-864f-43c9-972a-b5f2353389e6] - jobContext [419695287]
2023-06-09 10:27:29-651 | [JOB] INTCALC - Started
2023-06-09 10:27:29 default ERROR - c.n.b.g.r.c.i.ExecutionControllerImpl - Could not find program "CBACT04C" in the program registry.
2023-06-09 10:27:29 default ERROR - c.n.b.g.r.c.i.ExecutionControllerImpl - Could not find program "CBACT04C" in the program registry.
2023-06-09 10:27:29-760 | Program not found => not executed !
2023-06-09 10:27:29-761 | [STEP] STEP15 - Ended
2023-06-09 10:27:29-772 | [JOB] INTCALC - Ended
2023-06-09 10:27:29 default INFO - c.n.b.g.r.j.s.DefaultJobContext - Job [419695287] - starting final operation
2023-06-09 10:27:29 default INFO - c.n.b.g.r.j.s.DefaultJobContext - End of job [419695287]
2023-06-09 10:27:29 default INFO - c.n.b.g.r.s.ScriptControlTower - Removed jobExecutor [a65c2791-864f-43c9-972a-b5f2353389e6] from Script Control Tower.
2023-06-09 10:27:29 default INFO - c.n.b.g.r.s.ScriptControlTower - Remaining jobExecutors:0

```

### Note

Le chiamate sincrone devono essere riservate ai lavori di breve durata. I lavori con esecuzione prolungata dovrebbero invece essere avviati in modo asincrono (vedi endpoint dedicato di seguito).

## Avvia uno script in modo asincrono

- Metodi supportati: GET/POST
- Percorso: `/triggerscript/{scriptId:.+}`
- argomenti:
  - identificatore dello script da lanciare
  - opzionalmente: parametri da passare allo script, utilizzando i parametri di richiesta (visti come `aMap<String,String>`). I parametri forniti verranno aggiunti automaticamente a `https://docs.groovy-lang.org/latest/html/api/groovy/lang/Binding.html#bindings` dello script groovy richiamato.
- A differenza della modalità sincrona descritta in precedenza, l'endpoint non attende il completamento dell'esecuzione del job per inviare una risposta. L'esecuzione del processo viene avviata immediatamente, se è possibile trovare un thread disponibile in tal senso, e al chiamante viene inviata immediatamente una risposta con l'ID di esecuzione del lavoro, un identificatore univoco che rappresenta l'esecuzione del lavoro, che può essere utilizzato per interrogare lo stato di esecuzione del lavoro o forzare l'interruzione di un'esecuzione del lavoro che si suppone non funzioni correttamente. Il formato della risposta è:

```
Triggered script <script identifier> [unique job execution id] @ <date and time>
```

- Poiché l'esecuzione asincrona del processo si basa su un numero fisso limitato di thread, l'esecuzione del lavoro potrebbe non essere avviata se non viene trovato alcun thread disponibile. In tal caso, il messaggio restituito sarà piuttosto simile a:

```
Script [<script identifier>] NOT triggered - Thread limit reached (<actual thread limit>) - Please retry later or increase thread limit.
```

Vedi `settriggerthreadlimitendpoint` qui sotto per scoprire come aumentare il limite dei thread.

Risposta di esempio:

```
Triggered script INTCALC [d43cbf46-4255-4ce2-aac2-79137573a8b4] @ 06-12-2023 16:26:15
```

L'identificatore univoco di esecuzione del lavoro consente di recuperare rapidamente le voci di registro correlate nei log del server, se necessario. Viene utilizzato anche da diversi altri endpoint descritti di seguito.

Elenco degli script attivati

- Metodi supportati: GET
- Percorsi: `/triggeredscripsts/{status:.+}`, `/triggeredscripsts/{status:.+}/  
{namefilter}`
- Argomenti:
  - Status (obbligatorio): lo stato degli script attivati da recuperare. I valori possibili sono:
    - `all`: mostra tutti i dettagli sull'esecuzione del lavoro, indipendentemente dal fatto che i lavori siano ancora in esecuzione o meno.
    - `running`: mostra solo i dettagli dei lavori attualmente in esecuzione.
    - `done`: mostra i dettagli dei lavori solo per i lavori la cui esecuzione è terminata.
    - `killed`: mostra i dettagli dei lavori solo per i lavori la cui esecuzione è stata interrotta forzatamente utilizzando l'endpoint dedicato (vedi sotto).
    - `triggered`: mostra i dettagli dei lavori solo per i lavori che sono stati attivati ma non ancora avviati.
    - `failed`: mostra i dettagli dei lavori solo per i lavori la cui esecuzione è stata contrassegnata come fallita.

- `_namefilter` (opzionale) `_`: recupera solo le esecuzioni per l'identificatore di script specificato.
- Restituisce una raccolta di dettagli sulle esecuzioni dei lavori in formato JSON. Per ulteriori informazioni, consulta [Struttura dei messaggi Job Execution Details](#).

Risposta di esempio:

```
[
  {
    "scriptId": "INTCALC",
    "caller": "127.0.0.1",
    "identifier": "d43cbf46-4255-4ce2-aac2-79137573a8b4",
    "startTime": "06-12-2023 16:26:15",
    "endTime": "06-12-2023 16:26:15",
    "status": "DONE",
    "executionResult": "{ \"exitCode\": -1, \"stepName\": \"STEP15\", \"program\": \"CBACT04C\", \"status\": \"Error\" }",
    "executionMode": "ASYNCHRONOUS"
  }
]
```

Recupero dei dettagli sull'esecuzione del lavoro

- Metodo supportato: GET
- Percorso: `/getjobexecutioninfo/{jobexecutionid:.+}`
- Argomenti:
  - `jobexecutionid` (obbligatorio): l'identificatore univoco di esecuzione del lavoro per recuperare i dettagli di esecuzione del lavoro corrispondenti.
- Restituisce: una stringa JSON che rappresenta un singolo dettaglio sull'esecuzione del lavoro (vedi [Struttura dei messaggi Job Execution Details](#)) o una risposta vuota se non è stato possibile trovare dettagli sull'esecuzione del lavoro per l'identificatore specificato.

Elenco degli script avviati in modo asincrono che possono essere eliminati

- Metodo supportato: GET
- Percorso: `/killablescripts`

- Restituisce una raccolta di identificatori di esecuzione dei lavori che sono stati avviati in modo asincrono che sono ancora attualmente in esecuzione e che possono essere interrotti forzatamente (vedi l'endpoint di seguito). `/kill`

Elenco degli script avviati in modo sincrono che possono essere eliminati

- Metodo supportato: GET
- Percorso: `/killablesyncscripts`
- Restituisce una raccolta di identificatori di esecuzione dei lavori che sono stati avviati in modo sincrono, sono ancora attualmente in esecuzione e possono essere interrotti forzatamente (vedi l'endpoint di seguito).

Uccidere una determinata esecuzione di lavoro

- Metodo supportato: GET
- Percorso: `/kill/{identifier:.+}`
- argomento: identificatore di esecuzione del lavoro (obbligatorio): l'identificatore univoco di esecuzione del lavoro che indica che l'esecuzione del lavoro deve essere interrotta con la forza.
- Restituisce: un messaggio testuale che descrive in dettaglio l'esito del tentativo di interruzione dell'esecuzione del lavoro; il messaggio conterrà l'identificatore dello script, l'identificatore univoco dell'esecuzione del lavoro e la data e l'ora in cui si è verificato il completamento dell'esecuzione. Se non è possibile trovare alcuna esecuzione del processo in esecuzione per l'identificatore specificato, verrà invece restituito un messaggio di errore.

#### Warning

- Il runtime fa del suo meglio per interrompere correttamente l'esecuzione del job di destinazione. Pertanto, la risposta dall'endpoint `/kill` potrebbe impiegare un po' di tempo prima che arrivi al chiamante, poiché il runtime AWS Blu Age cercherà di ridurre al minimo l'impatto aziendale dell'interruzione del lavoro.
- Interrompere con la forza l'esecuzione di un lavoro non dovrebbe essere fatto alla leggera, in quanto potrebbe avere conseguenze aziendali dirette, tra cui la possibile perdita o il danneggiamento dei dati. Dovrebbe essere riservata ai casi in cui l'esecuzione di un determinato lavoro è andata storta e i mezzi per la correzione dei dati sono chiaramente identificati.

- La soppressione di un posto di lavoro dovrebbe portare a ulteriori indagini (analisi post mortem) per capire cosa è andato storto e adottare misure correttive adeguate.
- In ogni caso, il tentativo di terminare un processo in esecuzione verrà registrato nei log del server con messaggi a livello di avviso.

## Elenco dei checkpoint esistenti per la riavviabilità

La riavvio del lavoro si basa sulla capacità degli script di registrare i checkpoint per tracciare l'avanzamento dell'esecuzione del CheckpointRegistry lavoro. Se l'esecuzione di un job non riesce a terminare correttamente e i checkpoint di riavvio sono stati registrati, è sufficiente riavviare l'esecuzione del job dall'ultimo checkpoint registrato conosciuto (senza dover eseguire i passaggi precedenti al checkpoint).

- Metodo supportato: GET
- Percorso: `/restarts`
- Restituisce l'elenco dei punti di riavvio esistenti, che possono essere utilizzati per riavviare un lavoro la cui esecuzione non è avvenuta e non è terminata correttamente, come pagina html. Se nessuno script ha registrato alcun checkpoint, il contenuto della pagina sarà «Nessun checkpoint registrato»..

## Riavvio di un lavoro (in modo sincrono)

- Metodo supportato: GET
- Percorso: `/restart/{hashcode}`
- Argomenti: hashcode (intero - obbligatorio): riavvia l'esecuzione di un job precedentemente interrotto, utilizzando l'hashcode fornito come valore di checkpoint (vedi `/restartsendpoint` sopra per sapere come recuperare un valore di checkpoint valido).
- Restituzioni: vedi la descrizione del reso sopra. `script`

## Riavvio di un lavoro (in modo asincrono)

- Metodo supportato: GET
- Percorso: `/triggerrestart/{hashcode}`



- **Argomenti:** hashcode (intero - obbligatorio): riavvia l'esecuzione di un job precedentemente interrotto, utilizzando l'hashcode fornito come valore di checkpoint (vedi l'/restartsendpoint sopra per sapere come recuperare un valore di checkpoint valido).
- **Restituzioni:** vedi la descrizione del reso sopra. `triggerscript`

### Impostazione del limite di thread per le esecuzioni di lavori asincroni

L'esecuzione asincrona del processo si basa su un pool di thread dedicato nella JVM. Tale pool ha un limite fisso per quanto riguarda il numero di thread disponibili. L'utente ha la possibilità di regolare il limite in base alle capacità dell'host (numero di CPU, memoria disponibile, ecc...). Per impostazione predefinita, il limite di thread è impostato su 5 thread.

- **Metodo supportato:** GET
- **Percorso:** `/settriggerthreadlimit/{threadlimit:.+}`
- **Argomento (numero intero):** il nuovo limite di thread da applicare. Deve essere un numero intero strettamente positivo.
- **Restituisce un messaggio (String)** che indica il nuovo limite del thread e quello precedente, oppure un messaggio di errore se il valore limite del thread fornito non è valido (non è un numero intero strettamente positivo).

Risposta di esempio:

```
Set thread limit for Script Tower Control to 10 (previous value was 5)
```

### Conteggio delle esecuzioni di job attivate attualmente in esecuzione

- **Metodo supportato:** GET
- **Percorso:** `/countrunningtriggeredscrip`
- **Restituisce un messaggio** che indica il numero di processi in esecuzione avviati in modo asincrono e il limite di thread (ovvero il numero massimo di processi attivati che possono essere eseguiti contemporaneamente).

Risposta di esempio:

```
0 triggered script(s) running (limit =10)
```

### Note

Questo può essere usato per verificare, prima di avviare un lavoro, se il limite del thread non è stato raggiunto (il che impedirebbe l'avvio del processo).

Elimina le informazioni sulle esecuzioni dei job

Le informazioni sulle esecuzioni dei processi rimangono nella memoria del server finché il server è attivo. Potrebbe essere conveniente eliminare le informazioni più vecchie dalla memoria, poiché non sono più rilevanti; questo è lo scopo di questo endpoint.

- Metodo supportato: GET
- Percorso: `/purgejobinformation/{age:.+}`
- Argomenti: un valore intero strettamente positivo che rappresenta l'età in ore delle informazioni da eliminare.
- Restituisce un messaggio con le seguenti informazioni:
  - Nome del file di eliminazione in cui le informazioni eliminate sull'esecuzione del lavoro vengono archiviate a scopo di archiviazione.
  - Numero di informazioni sull'esecuzione del lavoro eliminate.
  - Numero di informazioni rimanenti sull'esecuzione del lavoro nel memo

## Metriche, endpoint

### JVM

Questo endpoint restituisce le metriche disponibili relative alla JVM.

- Metodo supportato: GET
- Percorso: `/metrics/jvm`
- Argomenti: nessuno
- Restituisce un messaggio con le seguenti informazioni:
  - `threadActiveCount`: numero di thread attivi.
  - `jvmMemoryUsed`: memoria utilizzata attivamente dalla Java Virtual Machine.
  - `jvmMemoryMax`: Memoria massima consentita per la Java Virtual Machine.

- `jvmMemoryFree`: Memoria disponibile non attualmente utilizzata dalla Java Virtual Machine.

## Sessione

Questo endpoint restituisce le metriche relative alle sessioni HTTP attualmente aperte.

- Metodo supportato: GET
- Percorso: `/metrics/session`
- Argomenti: nessuno
- Restituisce un messaggio con le seguenti informazioni:
  - `SessionCount`: numero di sessioni utente attive attualmente gestite dal server.

## Archiviazione

- Metodo supportato: GET
- Percorso: `/metrics/batch`
- Argomenti:
  - `startTimestamp` (opzionale, numero): timestamp iniziale per il filtraggio dei dati.
  - `EndTimestamp` (opzionale, numero): timestamp di fine per il filtraggio dei dati.
  - `pagina` (opzionale, numero): numero di pagina per l'impaginazione.
  - `PageSize` (opzionale, numero): numero di elementi per pagina nell'impaginazione.
- Restituisce un messaggio con le seguenti informazioni:
  - `contenuto`: Elenco delle metriche di esecuzione in batch.
  - `PageNumber`: numero di pagina corrente nell'impaginazione.
  - `PageSize`: numero di elementi visualizzati per pagina.
  - `TotalPages`: numero totale di pagine disponibili.
  - `numberOfElements`: Numero di articoli nella pagina corrente.
  - `last`: contrassegno booleano per l'ultima pagina.
  - `first`: bandiera booleana per la prima pagina.

## Transaction

- Metodo supportato: GET

- Percorso: `/metrics/transaction`
- Argomenti:
  - `startTimestamp` (opzionale, numero): timestamp iniziale per il filtraggio dei dati.
  - `EndTimestamp` (opzionale, numero): timestamp di fine per il filtraggio dei dati.
  - `pagina` (opzionale, numero): numero di pagina per l'impaginazione.
  - `PageSize` (opzionale, numero): numero di elementi per pagina nell'impaginazione.
- Restituisce un messaggio con le seguenti informazioni:
  - `contenuto`: Elenco delle metriche di esecuzione delle transazioni.
  - `PageNumber`: numero di pagina corrente nell'impaginazione.
  - `PageSize`: numero di elementi visualizzati per pagina.
  - `TotalPages`: numero totale di pagine disponibili.
  - `numberOfElements`: Numero di articoli nella pagina corrente.
  - `last`: contrassegno booleano per l'ultima pagina.
  - `first`: bandiera booleana per la prima pagina.

## Altri endpoint

Usa questi endpoint per elencare programmi o servizi registrati, scoprire lo stato di salute e gestire le transazioni JICS.

### Argomenti

- [Elencare i programmi registrati](#)
- [Elenco dei servizi registrati](#)
- [Health status \(Stato di integrità\)](#)
- [Elenco delle transazioni JICS disponibili](#)
- [Avvia una transazione JICS](#)
- [Avviare una transazione JICS \(alternativa\)](#)

### Elencare i programmi registrati

- Metodo supportato: GET
- Percorso: `/programs`

- Restituisce l'elenco dei programmi registrati, come pagina html. Ogni programma è designato dal suo identificatore principale. Nell'elenco vengono restituiti sia i programmi legacy modernizzati che i programmi di utilità (IDCAMS, IEBGENER, ecc...). Tieni presente che i programmi di utilità disponibili dipenderanno dalle applicazioni web di utilità che sono state distribuite sul tuo server tomcat. Ad esempio, i programmi di supporto delle utilità z/OS potrebbero non essere disponibili per gli asset iSeries modernizzati, in quanto non pertinenti.

#### Elenco dei servizi registrati

- Metodo supportato: GET
- Percorso: `/services`
- Restituisce l'elenco dei servizi di runtime registrati, come pagina html. I servizi forniti sono forniti dal runtime AWS Blu Age come utilità, che possono essere utilizzate ad esempio negli script groovy. I servizi di caricamento Blusam (per creare set di dati Blusam da set di dati legacy) rientrano in questa categoria.

#### Risposta di esempio:

```
<p>BluesamESDSFileLoader</p><p>BluesamKSDSFileLoader</p><p>BluesamRRDSFileLoader</p>
```

#### Health status (Stato di integrità)

- Metodo supportato: GET
- Percorso: `/`
- Restituisce un messaggio semplice che indica che l'applicazione gapwalk è attiva e funzionante ()  
`Jics application is running.`

#### Elenco delle transazioni JICS disponibili

- Metodo supportato: GET
- Percorso: `/transactions`
- Restituisce una pagina html che elenca tutte le transazioni JICS disponibili. Ciò ha senso solo per gli ambienti con elementi JICS (modernizzazione degli elementi CICS precedenti).

#### Risposta di esempio:

```
<p>INQ1</p><p>MENU</p><p>MNT2</p><p>ORD1</p><p>PRNT</p>
```

## Avvia una transazione JICS

- Metodi supportati: GET, POST
- Percorso: `/jicstransrunner/{jtrans: .+}`
- argomenti:
  - Identificatore di transazione JICS (stringa, obbligatorio): identificatore della transazione JICS da avviare (lunghezza massima 8 caratteri)
  - `<String, Object>` richiesto: dati di input aggiuntivi da passare alla transazione, come mappa. Il contenuto di questa mappa verrà utilizzato per alimentare la [COMMAREA](#) che verrà consumata dalla transazione JICS. La mappa può essere vuota se non sono richiesti dati per eseguire la transazione.
  - opzionale: voci di intestazioni Http, per personalizzare l'ambiente di esecuzione per una determinata transazione. Sono supportate le seguenti chiavi di intestazione:
    - `jics-channel`: Il nome del JICS CHANNEL che verrà utilizzato dal programma che verrà lanciato al momento dell'avvio della transazione.
    - `jics-container`: Il nome del JICS CONTAINER da utilizzare per l'avvio di questa transazione JICS.
    - `jics-startcode`: lo STARTCODE (stringa, fino a 2 caratteri) da utilizzare all'inizio della transazione JICS. Vedi [STARTCODE](#) per i valori possibili (sfoglia la pagina).
    - `jicxa-xid`: Lo XID (struttura X/Open transaction identifier XID) di una «transazione globale» ([XA](#)), [avviata dal chiamante, a](#) cui parteciperà l'attuale avvio della transazione JICS.
- Restituisce: una serializzazione `com.netfactive.bluage.gapwalk.rt.shared.web.TransactionResultBean` JSON, che rappresenta il risultato dell'avvio della transazione JICS.

Per ulteriori informazioni sui dettagli della struttura, vedere. [Struttura dei risultati del lancio della transazione](#)

## Avviare una transazione JICS (alternativa)

- metodi supportati: GET, POST
- percorso: `/jicstransaction/{jtrans: .+}`
- argomenti:

identificatore di transazione JICS (stringa, obbligatorio)

identificatore della transazione JICS da avviare (lunghezza massima di 8 caratteri)

richiesto: dati di input aggiuntivi da passare alla transazione, come mappa<String, Object>

Il contenuto di questa mappa verrà utilizzato per alimentare la [COMMAREA](#) che verrà consumata dalla transazione JICS. La mappa può essere vuota se non sono richiesti dati per eseguire la transazione.

opzionale: voci di intestazioni Http, per personalizzare l'ambiente di esecuzione per una determinata transazione.

Sono supportate le seguenti chiavi di intestazione:

- `jics-channel`: Il nome del JICS CHANNEL che verrà utilizzato dal programma che verrà lanciato al momento dell'avvio della transazione.
  - `jics-container`: Il nome del JICS CONTAINER da utilizzare per l'avvio di questa transazione JICS.
  - `jics-startcode`: lo STARTCODE (stringa, fino a 2 caratteri) da utilizzare all'inizio della transazione JICS. Per i valori possibili, vedi [STARTCODE](#) (sfoglia la pagina).
  - `jicxa-xid`: Lo XID (struttura X/Open transaction identifier XID) di una «transazione globale» ([XA](#)), [avviata dal chiamante, a](#) cui parteciperà l'avvio della transazione JICS corrente.
- restituisce: una serializzazione  
`com.netfective.bluage.gapwalk.rt.shared.web.RecordHolderBean` JSON, che rappresenta il risultato dell'avvio della transazione JICS. I dettagli della struttura sono disponibili in [Struttura dei risultati del record di avvio delle transazioni](#)

## Endpoint relativi a Job Queues

Job Queues è il supporto AWS Blu Age per il meccanismo di invio dei lavori AS400. Le Job Queue vengono utilizzate in AS400 per eseguire job su pool di thread specifici. Una coda di processi è definita da un nome e da un numero massimo di thread che corrisponde al numero massimo di programmi che possono essere eseguiti contemporaneamente su quella coda. Se in coda vengono inviati più lavori rispetto al numero massimo di thread, i lavori aspetteranno che un thread sia disponibile.

Per un elenco completo dello stato di un lavoro in coda, vedi. [Possibile stato di un lavoro in coda](#)

Le operazioni sulle code di lavoro vengono gestite tramite i seguenti endpoint dedicati. È possibile richiamare queste operazioni dall'URL dell'applicazione Gapwalk con il seguente URL principale: `http://server:port/gapwalk-application/jobqueue`

### Argomenti

- [Elenca le code disponibili](#)
- [Avvia o riavvia una coda di lavori](#)
- [Invia un lavoro per il lancio](#)
- [Elenca tutti i lavori pianificati](#)
- [Elenca tutti i lavori «in attesa»](#)
- [Elenca tutti i lavori attivi](#)
- [Elenca tutti i lavori in attesa di essere lanciati](#)
- [Rilascia tutti i lavori che sono «in attesa»](#)
- [Rilascia tutti i lavori che sono «in attesa» per un determinato nome di lavoro](#)
- [Rilascia un determinato lavoro per un numero di lavoro](#)

### Elenca le code disponibili

- Metodo supportato: GET
- Percorso: `list-queues`
- Restituisce l'elenco delle code disponibili insieme al loro stato, come elenco JSON di valori-chiave.

### Risposta di esempio:

```
{"Default":"STAND_BY","queue1":"STARTED","queue2":"STARTED"}
```

Gli stati possibili per una coda di lavoro sono:

#### STAND\_BY

la coda dei lavori è in attesa di essere avviata.

#### AVVIATA

la coda dei lavori è attiva e funzionante.



## UNKNOWN

lo stato della coda dei lavori non può essere determinato.

Avvia o riavvia una coda di lavori

- Metodo supportato: POST
- Percorso: `/restart/{name}`
- Argomento: il nome della coda da avviare/riavviare, come stringa - obbligatorio.
- L'endpoint non restituisce nulla ma si affida piuttosto allo stato http per indicare l'esito dell'operazione di avvio/riavvio:

HTTP 200

l'operazione di avvio/riavvio è andata bene: la coda di lavoro specificata è ora AVVIATA.

HTTP 404

la coda dei lavori non esiste.

HTTP 503

si è verificata un'eccezione durante il tentativo di avvio/riavvio (i log del server devono essere controllati per capire cosa è andato storto).

Invia un lavoro per il lancio

- Metodo supportato: POST
- Argomento: obbligatorio come corpo della richiesta, serializzazione JSON di un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` oggetto. Per ulteriori informazioni, consulta [Invia input per il lavoro](#).
- Restituisce: un JSON contenente l'originale `SubmitJobMessage` e un registro che indica se il lavoro è stato inviato o meno.

Elenca tutti i lavori pianificati

- Metodo supportato: GET
- Percorso: `list-jobs`

- Restituisce: un elenco di tutti i lavori pianificati, come stringa JSON. Per un esempio di risposta, vedere [Elenco delle risposte ai lavori pianificati](#).

#### Elenca tutti i lavori «in attesa»

- Metodo supportato: GET
- Percorso: `list-jobs-hold`
- Restituisce: un elenco di tutti i lavori pianificati, come stringa JSON. Per un esempio di risposta, vedere [Elenco delle risposte alle offerte di lavoro «in sospeso»](#).

#### Elenca tutti i lavori attivi

- Metodo supportato: GET
- Percorso: `list-jobs-active`
- Restituisce: un elenco di tutti i lavori con stato ACTIVE, come stringa JSON. La risposta ha una struttura simile a quella di [Elenco delle risposte ai lavori pianificati](#).

#### Elenca tutti i lavori in attesa di essere lanciati

- Metodo supportato: GET
- Percorso: `list-jobs-waiting`
- Restituisce: un elenco di tutti i lavori con stato EXECUTION\_WAIT (lavori in attesa che un thread sia disponibile per il lancio), come stringa JSON. La risposta è simile nella struttura a quella di [the section called “Elenco delle risposte ai lavori pianificati”](#)

#### Rilascia tutti i lavori che sono «in attesa»

- Metodo supportato: POST
- Percorso: `release-all`
- Restituisce: un messaggio che indica l'esito dell'operazione di tentativo di rilascio. Qui ci sono due casi possibili:
  - HTTP 200 e un messaggio «Tutti i lavori sono stati rilasciati con successo!» se tutti i lavori sono stati rilasciati correttamente.

- HTTP 503 e un messaggio «Jobs not released». Si è verificato un errore sconosciuto. Vedi il registro per maggiori dettagli» se qualcosa è andato storto con il tentativo di rilascio.

Rilascia tutti i lavori che sono «in attesa» per un determinato nome di lavoro

<job name, job number>Per un determinato nome di lavoro, è possibile inviare più lavori, con numeri di lavoro diversi (l'unicità di un lavoro eseguito è garantita da un paio di lavori). L'endpoint tenterà di rilasciare tutte le offerte di lavoro con il nome del lavoro specificato, che sono «in attesa».

- Metodo supportato: POST
- Percorso: `/release/{name}`
- Argomenti: il nome del lavoro da cercare, sotto forma di stringa. Obbligatorio.
- Restituisce: un messaggio che indica l'esito dell'operazione di tentativo di rilascio. Qui ci sono due casi possibili:
  - HTTP 200 e un messaggio «Jobs in group <name>(<number of released jobs>) rilasciato con successo!» i lavori sono stati rilasciati con successo.
  - HTTP 503 e un messaggio «Jobs in group <name>not released. Si è verificato un errore sconosciuto. Vedi il registro per maggiori dettagli» se qualcosa è andato storto con il tentativo di rilascio.

Rilascia un determinato lavoro per un numero di lavoro

<job name, job number>L' endpoint tenterà di rilasciare l'unica candidatura di lavoro «in attesa» per la coppia specificata.

- Metodo supportato: POST
- Percorso: `/release/{name}/{number}`
- Argomenti:

nome

il nome del lavoro da cercare, come stringa. Obbligatorio.

number

il numero del lavoro da cercare, come numero intero. Obbligatorio.

restituisce

un messaggio che indica l'esito dell'operazione di tentativo di rilascio. Ecco due casi possibili:

- HTTP 200 e un messaggio «» Job <name/number> rilasciato con successo!» se il lavoro è stato rilasciato con successo.
- HTTP 503 e un messaggio «Job <name/number> not released. Si è verificato un errore sconosciuto. Vedi il registro per maggiori dettagli» se qualcosa è andato storto con il tentativo di rilascio.

## Endpoint REST della Blusam Application Console

La Blusam Application Console è un'API progettata per semplificare la gestione dei set di dati VSAM modernizzati. Gli endpoint per l'applicazione web Blusam utilizzano il percorso root. /bac

Argomenti

- [Set di dati \(endpoint correlati\)](#).
- [Set di dati in blocco: endpoint correlati](#)
- [Registri](#)
- [Maschere](#)
- [Altro](#)
- [Utenti](#)

Set di dati (endpoint correlati).

Utilizza i seguenti endpoint per creare o gestire un set di dati specifico.

Argomenti

- [Crea un set di dati](#)
- [Caricamento di un file](#)
- [Carica un set di dati](#)
- [Caricare un set di dati da un bucket Amazon S3](#)
- [Esportazione di un set di dati in un bucket Amazon S3](#)
- [Cancellare un set di dati](#)
- [Eliminare un set di dati](#)

- [Conta i record del set di dati](#)

## Crea un set di dati

La creazione di un endpoint di set di dati consente di creare una definizione di set di dati e richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/createDataSet``
- Argomenti:

nome

(obbligatorio, stringa): il nome del set di dati.

type (tipo)

(obbligatorio, stringa): il tipo di set di dati. I valori possibili sono:ESDS,KSDS,RRDS.

Dimensione del record

(opzionale, stringa): dimensione massima di ogni record del set di dati.

Lunghezza fissa

(opzionale, booleano): indica se la lunghezza dei record è fissa.

compressione

(opzionale, booleano): indica se il set di dati è compresso.

CacheEnable

(opzionale, booleano): indica se la memorizzazione nella cache è abilitata per il set di dati.

Tasti alternativi

(opzionale, elenco di chiavi):

- offset (obbligatorio, numero)
  - lunghezza (richiesto, numero)
  - nome (richiesto, numero)
- Restituisce un file json che rappresenta il set di dati appena creato.

**Richiesta di esempio:**

```
POST /api/services/rest/bluesamservice/createDataSet
{
  "name": "DATASET",
  "checked": false,
  "records": [],
  "primaryKey": {
    "name": "PK"
  },
  "alternativeKeys": [
    {
      "offset": 10,
      "length": 10,
      "name": "ALTK_0"
    }
  ],
  "type": "ESDS",
  "recordSize": 10,
  "compression": true,
  "cacheEnable": true
}
```

**Risposta di esempio:**

```
{
  "dataSet": {
    "name": "DATASET",
    "checked": false,
    "nbRecords": 0,
    "keyLength": -1,
    "recordSize": 10,
    "compression": false,
    "fixLength": true,
    "type": "ESDS",
    "cacheEnable": false,
    "cacheWarmup": false,
    "cacheEviction": "100ms",
    "creationDate": 1686744961234,
    "modificationDate": 1686744961234,
    "records": [],
    "primaryKey": {
      "name": "PK",
```

```
    "offset": null,
    "length": null,
    "columns": null,
    "unique": true
  },
  "alternativeKeys": [
    {
      "offset": 10,
      "length": 10,
      "name": "ALTK_0"
    }
  ],
  "readLimit": 0,
  "readEncoding": null,
  "initCharacter": null,
  "defaultCharacter": null,
  "blankCharacter": null,
  "strictZoned": null,
  "decimalSeparator": null,
  "currencySign": null,
  "pictureCurrencySign": null
},
"message": null,
"result": true
}
```

## Caricamento di un file

Questo endpoint consente di caricare file sul server. Il file viene archiviato in una cartella temporanea che corrisponde a ciascun utente specifico. Questo endpoint deve essere utilizzato ogni volta che è necessario caricare un file. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/upload`
- Argomenti:

file

(obbligatorio, multipart/form-data): il file da caricare.

- Restituisce un valore booleano che riflette lo stato del caricamento

## Carica un set di dati

Una volta creata la definizione del set di dati utilizzando l'endpoint di creazione del set di dati descritto in precedenza, è possibile caricare i record associati al file caricato su un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/loadDataSet`
- Argomenti:

nome

(obbligatorio, stringa): il nome del set di dati.

- Restituisce lo stato della richiesta e del set di dati caricato.

## Caricare un set di dati da un bucket Amazon S3

Carica un set di dati utilizzando un file listcat da un bucket Amazon S3.

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/loadDataSetFromS3``
- Argomenti:

Elenca la posizione dei file CAT 3

(obbligatorio, stringa): la posizione Amazon S3 del file listcat.

Posizione DataSetFiles3

(obbligatorio, stringa): la posizione Amazon S3 del file del set di dati.

Regione

(obbligatorio, stringa): l'Amazon S3 Regione AWS in cui sono archiviati i file.

- Restituisce il set di dati appena creato

Richiesta di esempio:

```
/BAC/api/services/rest/bluesamservice/loadDataSetFromS3?region=us-east-1&listcatFileS3Location=s3://bucket-name/listcat.json&datasetFileS3Location=s3://bucket-name/dataset.DAT
```



## Esportazione di un set di dati in un bucket Amazon S3

Esporta un set di dati nel bucket Amazon S3 specificato.

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/exportDataSetToS3`

- Argomenti:

`s3Location`

(obbligatorio, stringa): la posizione Amazon S3 in cui esportare il set di dati.

`datasetName`

(obbligatorio, stringa): il nome del set di dati da esportare.

`Regione`

(obbligatorio, stringa): il Regione AWS bucket Amazon S3.

- Restituisce il set di dati esportato

Richiesta di esempio:

```
/BAC/api/services/rest/bluesamservice/exportDataSetToS3?region=eu-west-1&s3Location=s3://bucket-name/dump&datasetName=dataset
```

## Cancellare un set di dati

Cancella tutti i record da un set di dati. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/clearDataSet`

- Argomenti:

`nome`

(obbligatorio, stringa): il nome del set di dati da cancellare.

- Restituisce lo stato della richiesta.

## Eliminare un set di dati

Elimina la definizione e i record del set di dati. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/deleteDataSet`
- Argomenti:
  - nome
    - (obbligatorio, stringa): il nome del set di dati da eliminare.
- Restituisce lo stato della richiesta e del set di dati eliminato.

### Conta i record del set di dati

Questo endpoint restituisce il numero di record associati a un set di dati. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/countRecords`
- Argomenti:
  - nome
    - (obbligatorio, stringa): il nome del set di dati.
- Restituisce: il numero di record

### Set di dati in blocco: endpoint correlati

Utilizza i seguenti endpoint per creare o gestire più set di dati contemporaneamente.

#### Argomenti

- [Esporta set di dati](#)
- [Crea più set di dati](#)
- [Elenca tutti i set di dati](#)
- [Elenca direttamente tutti i set di dati](#)
- [Elimina tutti i set di dati](#)
- [Ottieni le definizioni dei set di dati dal file listcat](#)
- [Ottieni le definizioni dei set di dati dal file cat dell'elenco caricato](#)
- [Carica listcat dal file json](#)

## Esporta set di dati

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/exportDataSet`
- Argomenti:

nome

(obbligatorio, stringa): il nome del set di dati da eliminare.

datasetOutputFile

(opzionale, stringa): il percorso in cui archiviare il set di dati esportato sul server  
rdw

(opzionale, booleano): devono essere esportati i campi RDW.

- restituisce lo stato della richiesta e del file contenente il set di dati esportato.

## Crea più set di dati

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/createAllDataSets`
- Argomenti:

- Elenco dei set di dati

nome

(obbligatorio, stringa): il nome del set di dati.

type (tipo)

(obbligatorio, stringa): il tipo di set di dati. I valori possibili sono:ESDS,KSDS,RRDS.

Dimensione del record

(opzionale, stringa): dimensione massima di ogni record del set di dati.

Lunghezza fissa

(opzionale, booleano): indica se la lunghezza dei record è fissa.

compressione

(opzionale, booleano): indica se il set di dati è compresso.

## CacheEnable

(opzionale, booleano): indica se la memorizzazione nella cache è abilitata per il set di dati.

- Restituisce: lo stato della richiesta e il set di dati appena creato.

### Elenca tutti i set di dati

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/listDataSet`
- Argomenti: Nessuno
- Restituisce: lo stato della richiesta e l'elenco dei set di dati.

### Elenca direttamente tutti i set di dati

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/directListDataSet`
- Argomenti: Nessuno
- Restituisce: lo stato della richiesta e l'elenco dei set di dati.

### Elimina tutti i set di dati

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/removeAll`
- Argomenti: Nessuno
- Restituisce: un valore booleano che rappresenta lo stato della richiesta.

### Ottieni le definizioni dei set di dati dal file listcat

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/getDataSetsDefinitionFromListcat`
- Argomenti:

## paramFilePath

(obbligatorio, stringa): il percorso del file listcat.

- Restituisce: un elenco di set di dati

Ottieni le definizioni dei set di dati dal file cat dell'elenco caricato

- Metodi supportati: POST
- Percorso: ``/api/services/rest/bluesamservice/getDataSetsDefinitionFromUploadedListcat``
- Argomenti: Nessuno
- Restituisce: un elenco di set di dati

Carica listcat dal file json

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/loadListcatFromJsonFile`
- Argomenti:  
filePath

(obbligatorio, stringa): il percorso del file listcat.

- Restituisce: un elenco di set di dati

## Registri

Utilizza i seguenti endpoint per creare o gestire i record all'interno di un set di dati.

Argomenti

- [Creazione di un record](#)
- [Leggi un set di dati](#)
- [Eliminazione di un registro](#)
- [Aggiornare un record](#)
- [Salva un record](#)

## Creazione di un record

Questo endpoint consente di creare un nuovo record. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/createRecord`

- Argomenti:

set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

maschera

(obbligatorio, maschera): l'oggetto maschera.

- Restituisce lo stato della richiesta e il record creato.

## Leggi un set di dati

Questo endpoint consente di leggere un set di dati.

- Metodi supportati: GET
- Percorso: `/api/services/rest/crud/readDataSet`

- Argomenti:

set di dati

(obbligatorio, DataSet): l'oggetto del set di dati.

- Restituisce lo stato della richiesta e il set di dati con i record.

## Eliminazione di un registro

Questo endpoint consente di eliminare un record da un set di dati. Richiede l'autenticazione.

- Metodi supportati: DELETE
- Percorso: `/api/services/rest/crud/deleteRecord`

- Argomenti:

set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

## record

(obbligatorio, Record): il record da eliminare

- Restituisce lo stato dell'eliminazione.

## Aggiornare un record

Questo endpoint consente di aggiornare un record associato a un set di dati. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/updateRecord`
- Argomenti:

set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

record

(obbligatorio, Record): il record da aggiornare

- Restituisce lo stato della richiesta e il set di dati con i record.

## Salva un record

Questo endpoint consente di salvare un record in un set di dati e di utilizzare una maschera. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/saveRecord`
- Argomenti:

set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

record

(richiesto, Record): il record da salvare

- Restituisce lo stato della richiesta e il set di dati con i record.

## Maschere

Utilizza i seguenti endpoint per caricare o applicare maschere a un set di dati.

### Argomenti

- [Caricare le maschere](#)
- [Applica la maschera](#)
- [Applica il filtro maschera](#)

### Caricare le maschere

Questo endpoint consente di recuperare tutte le maschere associate a un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/loadMasks`
- Argomenti:  
set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

- Restituisce lo stato della richiesta e l'elenco delle maschere.

### Applica la maschera

Questo endpoint consente di applicare una maschera a un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/applyMask`
- Argomenti:  
set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

maschera

(richiesto, Mask): l'oggetto del set di dati



- Restituisce lo stato della richiesta e il set di dati con la maschera applicata.

## Applica il filtro maschera

Questo endpoint consente di applicare una maschera e un filtro a un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/crud/applyMaskFilter`
- Argomenti:  
set di dati

(obbligatorio, DataSet): l'oggetto del set di dati

maschera

(richiesto, Mask): l'oggetto del set di dati

- Restituisce lo stato della richiesta e il set di dati con la maschera e il filtro applicati.

## Altro

Utilizza i seguenti endpoint per gestire la cache di un set di dati o controllare le caratteristiche del set di dati

### Argomenti

- [Controlla la cache di riscaldamento](#)
- [Verifica che la cache sia abilitata](#)
- [Abilita la cache](#)
- [Controlla la cache Ram allocata](#)
- [Controlla la persistenza](#)
- [Controlla i tipi di set di dati supportati](#)
- [Controlla lo stato del server](#)

## Controlla la cache di riscaldamento

Verifica se la cache di riscaldamento è abilitata per un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: POST
- Percorso: `/api/services/rest/bluesamservice/warmupCache``
- Argomenti:  
nome  
  
(obbligatorio, stringa): il nome del set di dati.
- Restituisce: true se la cache di riscaldamento è abilitata e false in caso contrario.

#### Verifica che la cache sia abilitata

Verifica se la cache è abilitata per un set di dati specifico. Richiede l'autenticazione.

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/isEnableCache`
- Argomenti: Nessuno
- Restituisce true se la memorizzazione nella cache è abilitata.

#### Abilita la cache

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/enableDisableCache/{enable}`
- Argomenti:  
abilita  
  
(obbligatorio, booleano): se impostato su true, abiliterà la memorizzazione nella cache.
- Restituisce Nessuno

#### Controlla la cache Ram allocata

Questo endpoint consente di recuperare la memoria cache RAM allocata. Richiede l'autenticazione.

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/allocatedRamCache`
- Argomenti: Nessuno
- Restituisce: la dimensione della memoria come stringa

## Controlla la persistenza

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/persistence`
- Argomenti: Nessuno
- Restituisce: la persistenza usata come stringa

## Controlla i tipi di set di dati supportati

- Percorso: `/api/services/rest/bluesamservice/getDataSetTypes`
- Argomenti: Nessuno
- Restituisce: l'elenco dei tipi di set di dati supportati come elenco di stringhe.

## Controlla lo stato del server

- Metodi supportati: GET
- Percorso: `/api/services/rest/bluesamservice/serverIsUp`
- Argomenti: Nessuno
- Restituzioni: nessuna

## Utenti

Utilizza i seguenti endpoint per gestire le interazioni con gli utenti.

### Argomenti

- [Login](#)
- [Controlla l'account utente](#)
- [Accedere](#)
- [Elencare tutti gli utenti](#)
- [Disconnettersi](#)

### Login

- Metodi supportati: POST

- Percorso: `/api/services/security/servicelogin/login`
- Argomenti:
  - `username`  
(obbligatorio, stringa)
  - `password`  
(obbligatorio, stringa)
- Restituisce il nome utente e i ruoli dell'utente che ha effettuato l'accesso

#### Risposta di esempio

```
{"login":"some-user","roles":[{"id":0,"roleName":"ROLE_ADMIN"}]}
```

#### Controlla l'account utente

- Metodi supportati: POST
- Percorso: `/api/services/security/servicelogin/hasAccount`
- Argomenti: Nessuno
- Restituisce: true se l'utente ha già effettuato l'accesso

#### Accedere

- Metodi supportati: POST
- Percorso: `/api/services/security/servicelogin/recorduser`
- Argomenti: Nessuno
- Restituisce: true se l'utente ha già effettuato l'accesso

#### Elencare tutti gli utenti

- Metodi supportati: GET
- Percorso: `/api/services/security/servicelogin/listusers`
- Argomenti: Nessuno
- Restituisce: l'elenco di tutti gli utenti

## Disconnettersi

- Metodi supportati: POST
- Percorso: `/api/services/security/servicelogout/logout`
- Argomenti: Nessuno
- Restituisce: true se l'utente è stato disconnesso con successo.

## Console applicativa JICS

Il componente JICS è il supporto AWS Blu Age per la modernizzazione delle risorse CICS legacy. L'applicazione web JICS Application Console è dedicata all'amministrazione delle risorse JICS. I seguenti endpoint consentono di eseguire le attività di amministrazione senza dover interagire con l'interfaccia utente JAC. Ogni volta che un endpoint richiede l'autenticazione, la richiesta dovrà includere i dettagli di autenticazione (nome utente/password in genere, come richiesto dall'autenticazione di base). Gli endpoint per l'applicazione Web JICS Application Console utilizzano il percorso root. `/jac/`

### Argomenti

- [Gestione delle risorse JICS](#)
- [Endpoint di gestione degli utenti JAC](#)

## Gestione delle risorse JICS

Tutti i seguenti endpoint sono correlati alla gestione delle risorse JICS, che consente agli amministratori JICS di gestire le risorse su base giornaliera.

### Argomenti

- [Elenca ELENCHI e GRUPPI JICS](#)
- [Elenca ICS GROUPS](#)
- [Elenca JICS GROUPS per un determinato ELENCO](#)
- [ELENCA le risorse JICS per un determinato GRUPPO](#)
- [ELENCA le risorse JICS per un determinato GRUPPO \(alternativa utilizzando un nome\)](#)
- [Modifica dei GRUPPI proprietari di diverse LISTE](#)
- [Eliminare un ELENCO](#)

- [Eliminare un GRUPPO](#)
- [Eliminare una TRANSAZIONE](#)
- [Eliminare un PROGRAMMA](#)
- [Eliminare un FILE](#)
- [Elimina un TDQUEUE](#)
- [Elimina un TSMODEL](#)
- [Crea un ELENCO](#)
- [Creare un GRUPPO](#)
- [Considerazioni comuni sulla creazione di RISORSE](#)
- [Crea una TRANSAZIONE](#)
- [Crea un PROGRAMMA](#)
- [Crea un FILE](#)
- [Crea un TDQUEUE](#)
- [Crea un TSMODEL](#)
- [Aggiorna un ELENCO](#)
- [Aggiorna un GRUPPO](#)
- [Considerazioni comuni sull'aggiornamento delle RISORSE](#)
- [Aggiorna una TRANSAZIONE](#)
- [Aggiorna un PROGRAMMA](#)
- [Aggiornare un FILE](#)
- [Aggiorna un TDQUEUE](#)
- [Aggiorna un TSMODEL](#)

## Elenca ELENCHI e GRUPPI JICS

LIST e GROUPS sono le principali risorse container proprietarie all'interno del componente JICS. Tutte le risorse JICS devono appartenere a un GRUPPO. I gruppi possono appartenere alle LISTE, ma ciò non è obbligatorio. Le LISTE potrebbero anche non esistere in un determinato ambiente JICS, ma la maggior parte delle volte le LISTE servono a fornire un ulteriore livello di organizzazione delle risorse. Per ulteriori informazioni sull'organizzazione delle risorse CICS, vedere Risorse [CICS](#).

- Metodo supportato: GET
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/listJicsListsAndGroups`
- Restituisce: un elenco di JicsContainer oggetti serializzati, sia LISTS che GROUPS, in formato JSON.

Risposta di esempio:

```
[
  {
    "name": "Resources",
    "children": [
      {
        "jacType": "JACList",
        "name": "MURACHS",
        "isActive": true,
        "children": [
          {
            "jacType": "JACGroup",
            "name": "MURACHS",
            "isActive": true,
            "children": []
          }
        ]
      },
      {
        "jacType": "JACGroup",
        "name": "TEST",
        "isActive": true,
        "children": []
      }
    ],
    "isExpanded": true
  }
]
```

## Elenca ICS GROUPS

- Metodo supportato: GET
- Richiede l'autenticazione

- Percorso: `/api/services/rest/jicsservice/listJicsGroups`
- Restituisce: un elenco di JicsContainer oggetti serializzati (GROUPS) in formato JSON. I GROUPS vengono restituiti senza le relative informazioni LIST.

Risposta di esempio:

```
[
  {
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  },
  {
    "jacType": "JACGroup",
    "name": "TEST",
    "isActive": true,
    "children": []
  }
]
```

Elenca JICS GROUPS per un determinato ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/listGroupsForList`
- Argomenti: un payload JSON, che rappresenta la JICS LIST di cui stiamo cercando i GROUPS. Questa è la serializzazione JSON di un oggetto.  
`com.netfactive.bluage.jac.entities.JACList`

Richiesta di esempio:

```
{
  "jacType": "JACList",
  "name": "MURACHS",
  "isActive": true
}
```

- Restituisce: un elenco di JicsContainer oggetti serializzati (GROUPS) in formato JSON, che sono allegati alla lista specificata. I GROUPS vengono restituiti senza le relative informazioni LIST.



**Risposta di esempio:**

```
[
  {
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  }
]
```

**ELENCA le risorse JICS per un determinato GRUPPO**

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/listResourcesForGroup`
- Argomenti: un payload JSON, che rappresenta il JICS GROUP di cui stiamo cercando le risorse. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.JACGroup` Non è necessario specificare tutti i campi per il GRUPPO, ma il nome è obbligatorio.

**Richiesta di esempio:**

```
{
  "jacType": "JACGroup",
  "name": "MURACHS",
  "isActive": true
}
```

- Restituisce: un elenco di `JicsResource` oggetti serializzati, di proprietà del gruppo specificato. Gli oggetti vengono restituiti senza un ordine particolare e sono di diversi tipi (PROGRAM, TRANSACTION, FILE, ecc...).

**ELENCA le risorse JICS per un determinato GRUPPO (alternativa utilizzando un nome)**

- Metodo supportato: POST
- Richiede l'autenticazione

- Percorso: `/api/services/rest/jicsservice/listResourcesForGroupName`
- Argomenti: il nome del GRUPPO che possiede le risorse che stiamo cercando.
- Restituisce: un elenco di JicsResource oggetti serializzati, di proprietà del gruppo specificato. Gli oggetti vengono restituiti senza un ordine particolare e sono di diversi tipi (PROGRAM, TRANSACTION, FILE, ecc...)

### Modifica dei GRUPPI proprietari di diverse LISTE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/editGroupsList`
- Argomenti: una rappresentazione JSON di una raccolta di LISTE con GRUPPI di bambini;

Richiesta di esempio:

```
[
  {
    "jacType": "JACList",
    "name": "MURACHS",
    "isActive": true,
    "children": [
      {
        "jacType": "JACGroup",
        "name": "MURACHS",
        "isActive": true,
        "children": []
      },
      {
        "jacType": "JACGroup",
        "name": "TEST",
        "isActive": true,
        "children": []
      }
    ]
  }
]
```

Prima di questa modifica, solo il gruppo denominato «MURACHS» apparteneva alla LISTA denominata «MURACHS». Con questa modifica, «aggiungiamo» il gruppo denominato «TEST» alla LISTA denominata «MURACHS».

- Restituisce un valore booleano. Se il valore è 'true', le modifiche a LISTS sono state mantenute correttamente nell'archivio JICS sottostante.

#### Eliminare un ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteList`
- Argomenti: un payload JSON, che rappresenta l'ELENCO JICS da eliminare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACList`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione di LIST è stata eseguita correttamente sulla memoria JICS sottostante.

#### Eliminare un GRUPPO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteGroup`
- Argomenti: un payload JSON, che rappresenta il JICS GROUP da eliminare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACGroup`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione GROUP è stata eseguita correttamente sullo storage JICS sottostante.

#### Eliminare una TRANSAZIONE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteTransaction`

- Argomenti: un payload JSON, che rappresenta la transazione JICS da eliminare. Questa è la serializzazione JSON di un oggetto.  
`com.netfactive.bluage.jac.entities.JACTransaction`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione delle TRANSAZIONI è stata eseguita correttamente sullo storage JICS sottostante.

### Eliminare un PROGRAMMA

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteProgram`
- Argomenti: un payload JSON, che rappresenta il programma JICS da eliminare. Questa è la serializzazione JSON di un oggetto. `com.netfactive.bluage.jac.entities.JACProgram`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione del PROGRAMMA è stata eseguita correttamente sulla memoria JICS sottostante.

### Eliminare un FILE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteFile`
- Argomenti: un payload JSON, che rappresenta il file JICS da eliminare. Questa è la serializzazione JSON di un oggetto. `com.netfactive.bluage.jac.entities.JACFile`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione dei FILE è stata eseguita correttamente sulla memoria JICS sottostante.

### Elimina un TDQUEUE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteTDQueue`
- Argomenti: un payload JSON, che rappresenta il JICS TDQUEUE da eliminare. Questa è la serializzazione JSON di un oggetto `com.netfactive.bluage.jac.entities.JACTDQueue``.

- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione di TDQUEUE è stata eseguita correttamente sulla memoria JICS sottostante.

### Elimina un TSMODEL

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/deleteTSModel`
- Argomenti: un payload JSON, che rappresenta il JICS TSMODEL da eliminare. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.JACTSModel``.
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione di TSMODEL è stata eseguita correttamente sulla memoria JICS sottostante.

### Crea un ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createList`
- Argomenti: un payload JSON, che rappresenta la LISTA JICS da creare. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.Jaclist``.
- Restituisce un valore booleano. Se il valore è 'true', l'ELENCO è stato creato correttamente nell'archivio JICS sottostante.

#### Note

La LIST verrà sempre creata vuota. Allegare GROUPS alla LIST richiederà un'altra operazione.

### Creare un GRUPPO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createGroup`

- Argomenti: un payload JSON, che rappresenta il JICS GROUP da creare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACGroup`
- Restituisce un valore booleano. Se il valore è 'true', il GROUP è stato creato correttamente nell'archivio JICS sottostante.

#### Note

Il GROUP verrà sempre creato vuoto. L'associazione di RESOURCES al GROUP richiederà operazioni aggiuntive (la creazione di risorse le collegherà automaticamente a un determinato GRUPPO).

## Considerazioni comuni sulla creazione di RISORSE

Tutti i seguenti endpoint sono correlati alla creazione di JICS RESOURCES e condividono alcuni vincoli comuni: nel payload della richiesta da inviare all'endpoint, il campo deve essere valutato.

`groupName`

Vincolo di proprietà del GRUPPO:

Nessuna risorsa può essere creata senza essere collegata a un gruppo esistente e l'endpoint utilizza `GroupName` per recuperare il gruppo a cui verrà collegata questa risorsa. `groupNameDeve` puntare al nome di un GROUP esistente. Verrà inviato un messaggio di errore con HTTP STATUS 400 se non punta a un gruppo esistente nell'archivio JICS sottostante. `groupName`

Vincolo di unicità all'interno di un GRUPPO:

Una determinata risorsa con un determinato nome deve essere unica all'interno di un determinato gruppo. Il controllo dell'unicità verrà eseguito da ciascun endpoint di creazione di risorse. Se il payload specificato non rispetta il vincolo di unicità, l'endpoint invierà una risposta con HTTP STATUS 400 (BAD REQUEST). Vedi la risposta di esempio riportata di seguito.

Payload di esempio: proviamo a creare la transazione «ARIT» nel gruppo «TEST», ma una transazione con quel nome esiste già in questo GRUPPO.

```
{
  "jacType": "JACTransaction",
  "name": "ARIT",
```

```
"groupName":"TEST",
"isActive":true
}
```

Abbiamo ricevuto la seguente risposta di errore:

```
{
  "timestamp": 1686759054510,
  "status": 400,
  "error": "Bad Request",
  "path": "/jac/api/services/rest/jicsservice/createTransaction"
}
```

L'ispezione dei log dei server confermerà l'origine del problema:

```
2023-06-14 18:10:54 default          TRACE - o.s.w.m.HandlerMethod
    - Arguments: [java.lang.IllegalArgumentException: Transaction already
present in the group, org.springframework.security.web.header.HeaderWriterFilter
$HeaderWriterResponse@e34f6b8]
2023-06-14 18:10:54 default          ERROR - c.n.b.j.a.WebConfig          -
400
java.lang.IllegalArgumentException: Transaction already present in the group
at
com.netfactive.bluage.jac.server.services.rest.impl.JicsServiceImpl.createElement(JicsServiceI
```

## Crea una TRANSAZIONE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createTransaction`
- Argomenti: un payload JSON, che rappresenta la TRANSAZIONE JICS da creare. Questa è la serializzazione JSON di un oggetto `com.netfactive.bluage.jac.entities.JACTransaction`
- Restituisce un valore booleano. Se il valore è 'true', la TRANSAZIONE è stata creata correttamente nell'archivio JICS sottostante.

## Crea un PROGRAMMA

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createProgram`
- Argomenti: un payload JSON, che rappresenta il PROGRAMMA JICS da creare. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.JACProgram`
- Restituisce un valore booleano. Se il valore è 'true', il PROGRAMMA è stato creato correttamente nella memoria JICS sottostante.

## Crea un FILE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createFile`
- Argomenti: un payload JSON, che rappresenta il FILE JICS da creare. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.JacFile``.
- Restituisce un valore booleano. Se il valore è 'true', il FILE è stato creato correttamente nell'archivio JICS sottostante.

## Crea un TDQUEUE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/createTDQueue`
- Argomenti: un payload JSON, che rappresenta il JICS TDQUEUE da creare. Questa è la serializzazione JSON di un oggetto `com.netfective.bluage.jac.entities.JACTDQueue``.
- Restituisce un valore booleano. Se il valore è 'true', TDQUEUE è stato creato correttamente nella memoria JICS sottostante.

## Crea un TSMODEL

- Metodo supportato: POST
- Richiede l'autenticazione



- Percorso: `/api/services/rest/jicsservice/createTSMoDel`
- Argomenti: un payload JSON, che rappresenta il JICS TSMODEL da creare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACTSMoDel`
- Restituisce un valore booleano. Se il valore è 'true', il TSMODEL è stato creato correttamente nella memoria JICS sottostante.

## Aggiorna un ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateList`
- Argomenti: un payload JSON, che rappresenta l'ELENCO JICS da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACList Non` è necessario fornire i figli del LIST, il meccanismo di aggiornamento LIST non ne terrà conto.
- Restituisce un valore booleano. Se il valore è 'true', l'ELENCO è stato aggiornato correttamente nell'archivio JICS sottostante.

L'aggiornamento del flag LIST 'isActive' si propagerà a tutti gli elementi di proprietà della LIST, ovvero a tutti i GRUPPI di proprietà della LIST e a tutte le RISORSE di proprietà di tali GRUPPI. Questo è un modo conveniente per disattivare molte risorse con un'unica operazione, su più GRUPPI.

## Aggiorna un GRUPPO

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateGroup`
- Argomenti: un payload JSON, che rappresenta il JICS GROUP da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACGroup` Non è necessario fornire i figli del GROUP, il meccanismo di aggiornamento GROUP non ne terrà conto.
- Restituisce un valore booleano. Se il valore è 'true', il GROUP è stato aggiornato correttamente nell'archivio JICS sottostante.

**Note**

L'aggiornamento del flag GROUP 'isActive' si propagherà a tutti gli elementi di proprietà del GROUP, ovvero a tutte le RISORSE di proprietà del GRUPPO. Questo è un modo conveniente per disattivare molte risorse con una singola operazione all'interno di un determinato GRUPPO.

**Considerazioni comuni sull'aggiornamento delle RISORSE**

Tutti i seguenti endpoint riguardano l'aggiornamento di JICS RESOURCES. Utilizzando il groupName campo, è possibile modificare il GROUP proprietario di qualsiasi RISORSA JICS, a condizione che il valore del campo punti a un GROUP esistente nell'archivio JIS sottostante (in caso contrario, si riceverà una risposta BAD REQUEST (HTTP STATUS 400) dall'endpoint).

**Aggiorna una TRANSAZIONE**

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateTransaction`
- Argomenti: un payload JSON, che rappresenta la TRANSAZIONE JICS da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACTransaction`
- Restituisce un valore booleano. Se il valore è 'true', la TRANSAZIONE è stata aggiornata correttamente nell'archivio JICS sottostante.

**Aggiorna un PROGRAMMA**

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateProgram`
- Argomenti: un payload JSON, che rappresenta il PROGRAMMA JICS da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACProgram`
- Restituisce un valore booleano. Se il valore è 'true', il PROGRAMMA è stato aggiornato correttamente nella memoria JICS sottostante.

## Aggiornare un FILE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateFile`
- Argomenti: un payload JSON, che rappresenta il FILE JICS da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACFile`
- Restituisce un valore booleano. Se il valore è 'true', il FILE è stato aggiornato correttamente nell'archivio JICS sottostante.

## Aggiorna un TDQUEUE

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateTDQueue`
- Argomenti: un payload JSON, che rappresenta il JICS TDQUEUE da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACTDQueue`
- Restituisce un valore booleano. Se il valore è 'true', TDQueue è stato aggiornato correttamente nella memoria JICS sottostante.

## Aggiorna un TSMODEL

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/updateTSModel`
- Argomenti: un payload JSON, che rappresenta il JICS TSMODEL da aggiornare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.JACTSModel`
- Restituisce un valore booleano. Se il valore è 'true', TSMODEL è stato aggiornato correttamente nella memoria JICS sottostante.

## Endpoint di gestione degli utenti JAC

### Argomenti

- [Eliminazione di un utente](#)

- [Registrazione di un utente](#)
- [Verifica se nel sistema esiste almeno un utente](#)
- [Registrazione di un nuovo utente](#)
- [Elenco degli utenti](#)
- [Elenco degli utenti](#)
- [Disconnette l'utente corrente](#)
- [Stato di salute del server Jics](#)

## Eliminazione di un utente

- Metodo supportato: POST
- Richiede l'autenticazione e i diritti di amministratore
- Percorso: `/api/services/security/servicelogin/deleteuser`
- Argomenti: la serializzazione JSON di un `com.netfactive.bluage.jac.entities.SignOn` oggetto, che rappresenta l'utente da rimuovere dall'archivio.
- Restituisce un valore booleano. Se il valore è 'true', l'utente è stato rimosso correttamente dal sistema JICS.

### Note

Questa azione non può essere annullata, l'utente eliminato non sarà in grado di connettersi nuovamente all'applicazione JAC. Utilizza questa soluzione con cautela.

## Registrazione di un utente

- Metodo supportato: POST
- Richiede l'autenticazione e i diritti di amministratore
- Percorso: `/api/services/security/servicelogin/login`
- Restituisce la serializzazione JSON di un `com.netfactive.bluage.jac.entities.SignOn` oggetto, che rappresenta l'utente le cui credenziali sono fornite nella richiesta corrente. La password è nascosta alla vista nell'oggetto restituito. I ruoli assegnati all'utente vengono elencati.

## Risposta di esempio:

```
{
  "login": "sadmin",
  "password": null,
  "roles": [
    {
      "id": 0,
      "roleName": "ROLE_SUPER_ADMIN"
    }
  ]
}
```

Verifica se nel sistema esiste almeno un utente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogin/hasAccount`
- Restituisce un valore booleano, il cui valore è vero se almeno un utente, diverso dall'utente super amministratore predefinito, è stato creato nel sistema JICS.

Registrazione di un nuovo utente

- Metodo supportato: POST
- Richiede l'autenticazione e i diritti di amministratore
- Percorso: `/api/services/security/servicelogin/recorduser`
- Argomenti: la serializzazione JSON di un `com.netfactive.bluage.jac.entities.SignOn` oggetto, che rappresenta l'utente da aggiungere allo storage. I ruoli per l'utente devono essere definiti, altrimenti l'utente potrebbe non essere in grado di utilizzare la struttura e gli endpoint JAC.

Richiesta di esempio:

```
{
  "login": "simpleuser",
  "password": "simpleuser",
  "roles": [
    {
      "id": 2,
      "roleName": "ROLE_USER"
    }
  ]
}
```

```
}
```

È possibile utilizzare solo i seguenti ruoli durante la registrazione di un nuovo utente:

- **ROLE\_ADMIN**: può gestire le risorse e gli utenti JICS.
- **ROLE\_USER**: può gestire le risorse JICS ma non gli utenti.

#### Elenco degli utenti

- Metodo supportato: GET
- Richiede l'autenticazione e i diritti di amministratore
- Percorso: `/api/services/security/servicelogin/listusers`
- Restituisce un elenco di `com.netfactive.bluage.jac.entities.SignOn`, serializzato come JSON.

#### Elenco degli utenti

- Metodo supportato: GET
- Richiede l'autenticazione e i diritti di amministratore
- Percorso: `/api/services/security/servicelogin/listusers`
- Restituisce un elenco di `com.netfactive.bluage.jac.entities.SignOn`, serializzato come JSON.

#### Disconnette l'utente corrente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogout/logout`
- Restituisce il seguente messaggio JSON: `«{" success":true}»` se il logout dell'utente corrente è riuscito (la sessione HTTP correlata verrà invalidata).

#### Stato di salute del server Jics

- Metodo supportato: GET
- Percorso: `/api/services/rest/jicsserver/serverIsUp`
- Indica che il server JICS è attivo e funzionante, se si riceve una risposta con HTTP STATUS 200.

## Strutture dati

Questa sezione descrive i dettagli delle varie strutture di dati.

### Argomenti

- [Struttura dei messaggi Job Execution Details](#)
- [Struttura dei risultati del lancio della transazione](#)
- [Struttura dei risultati del record di avvio delle transazioni](#)
- [Possibile stato di un lavoro in coda](#)
- [Invia input per il lavoro](#)
- [Elenco delle risposte ai lavori pianificati](#)
- [Elenco delle risposte alle offerte di lavoro «in sospeso»](#)

### Struttura dei messaggi Job Execution Details

I dettagli di esecuzione di ogni job avranno i seguenti campi:

#### ScriptID

l'identificatore dello script chiamato.

#### chiamante

indirizzo IP del chiamante.

#### identificatore

identificatore univoco di esecuzione del lavoro.

#### startTime

data e ora di inizio dell'esecuzione del lavoro.

#### endTime

data e ora in cui è terminata l'esecuzione del lavoro.

#### status

uno stato per l'esecuzione del lavoro. Un valore possibile tra:

- DONE: l'esecuzione del lavoro è terminata normalmente.

- TRIGGERED: esecuzione del processo attivata ma non ancora avviata.
- RUNNING: l'esecuzione del processo è in esecuzione.
- KILLED: l'esecuzione del lavoro è stata interrotta.
- FAILED: l'esecuzione del processo non è riuscita.

## Risultato dell'esecuzione

un messaggio per riassumere il risultato dell'esecuzione del lavoro. Questo messaggio può essere un messaggio semplice se l'esecuzione del lavoro non è ancora terminata o una struttura JSON con i seguenti campi:

- ExitCode: codice di uscita numerico; i valori negativi indicano situazioni di errore.
- programma: ultimo programma lanciato dal job.
- status: un valore possibile tra:
  - Error: quando exitCode = -1; ciò corrisponde a un errore (tecnico) che si verifica durante l'esecuzione del lavoro.
  - Failed: when exitcode = -2; Ciò corrisponde a un errore che si verifica durante l'esecuzione di un programma di servizio (come una situazione ABEND).
  - Succeeded: quando ExitCode >= 0;
- StepName: nome dell'ultimo passaggio eseguito nel job.

## Modalità di esecuzione

SINCRONO o ASINCRONO, a seconda del modo in cui il lavoro è stato avviato.

## Output di esempio:

```
{
  "scriptId": "INTCALC",
  "caller": "127.0.0.1",
  "identifier": "97d410be-efa7-4bd3-b7b9-d080e5769771",
  "startTime": "06-09-2023 11:42:41",
  "endTime": "06-09-2023 11:42:42",
  "status": "DONE",
  "executionResult": "{ \"exitCode\": -1, \"stepName\": \"STEP15\", \"program\": \"CBACT04C\", \"status\": \"Error\" }",
  "executionMode": "ASYNCHRONOUS"
}
```



## Struttura dei risultati del lancio della transazione

La struttura potrebbe contenere i seguenti campi:

### Risultato

una stringa che rappresenta il risultato dell'esecuzione della transazione. I valori possibili sono:

- **Success**: l'esecuzione della transazione è avvenuta correttamente.
- **Failure**: l'esecuzione della transazione non è riuscita a terminare correttamente, sono stati riscontrati alcuni problemi.

### - compara

una stringa che rappresenta il valore finale di COMMAREA, come matrice di byte con codifica byte64. Potrebbe essere una stringa vuota.

### ContainerRecord

(opzionale) una stringa che rappresenta il contenuto del record di CONTAINER come array di byte con codifica byte64.

### Descrizione del server

Può contenere informazioni sul server che ha fornito la richiesta (a scopo di debug). Potrebbe essere una stringa vuota.

### Un codice di fine

(opzionale) se il programma a cui fa riferimento la transazione avviata è annullato, il valore del codice abend verrà restituito come stringa in questo campo.

Esempi di risposte:

### Riuscito

```
{
  "outCome": "Success",
  "commarea": "",
  "serverDescription": ""
}
```

### Errore

```
{
  "outcome": "Failure",
  "commarea": "",
  "serverDescription": "",
  "abendCode": "AEIA"
}
```

## Struttura dei risultati del record di avvio delle transazioni

La struttura potrebbe contenere i seguenti campi:

### Contenuto del record

una stringa che rappresenta il contenuto del record di COMMAREA come un array di byte con codifica byte64.

### ContainerRecord

una stringa che rappresenta il contenuto del record del CONTAINER come array di byte codificato in byte64.

### Descrizione del server

Può contenere informazioni sul server che ha fornito la richiesta (a scopo di debug). Potrebbe essere una stringa vuota.

Esempi di risposte:

### Riuscito

```
{
  "recordContent": "",
  "serverDescription": ""
}
```

## Possibile stato di un lavoro in coda

In una coda, i lavori possono avere il seguente stato:

### ACTIVE

Il processo è attualmente in esecuzione sulla coda.

## ESECUZIONE\_ATTESA

Il lavoro è in attesa che un thread sia disponibile.

## SCHEDULED

L'esecuzione dei lavori è pianificata in una data e ora specifiche.

## HOLD

Job è in attesa di essere rilasciato prima di essere eseguito.

## COMPLETED

Job è stato eseguito con successo.

## Non riuscito

L'esecuzione del Job non è riuscita.

## UNKNOWN

Lo stato è sconosciuto.

## Invia input per il lavoro

L'input di invio del lavoro è la serializzazione JSON di un `com.netfactive.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` oggetto. L'input di esempio riportato di seguito mostra tutti i campi relativi a tale bean.

Esempio di input:

```
{
  "messageQueueName":null,
  "scheduleDate":null,
  "scheduleTime":null,
  "programName":"PTA0044",
  "programParams":
  {"wmind":"B"},
  "localDataAreaValue":"",
  "userName":"USER1",
  "jobName":"PTA0044",
  "jobNumber":9,
  "jobPriority":5,
```

```
"executionDate":"20181231",  
"jobQueue":"queue1",  
"jobOnHold":false  
}
```

## Numero del lavoro

se il numero del lavoro è 0, il numero del lavoro verrà generato automaticamente utilizzando il numero successivo nella sequenza numerica del lavoro. Tale valore deve essere impostato su 0 (tranne che a scopo di test).

## Priorità del lavoro

La priorità del lavoro predefinita in AS400 è 5. L'intervallo valido è 0-9, dove 0 è la priorità più alta.

## jobOnHold

Se un lavoro viene inviato in sospenso, non verrà eseguito immediatamente ma solo quando qualcuno lo «rilascia». Un job può essere rilasciato utilizzando l'API REST (/release o /release-all).

## ScheduleDate e ScheduleTime

Se questi valori non sono nulli, il lavoro verrà eseguito alla data e all'ora specificate.

## Data

può essere fornito con il formato MMDDYy o ddMMyyyy (la dimensione dell'input determinerà il formato utilizzato)

## Orario

può essere fornito con il formato HHmm o hHMMss (la dimensione dell'input determinerà il formato utilizzato)

## Parametri del programma

questo verrà passato al programma come mappa.

## Elenco delle risposte ai lavori pianificati

Questa è la struttura dell'endpoint list-jobs job queue. Tieni presente che il messaggio di invio del lavoro utilizzato per inviare quel lavoro fa parte della risposta. Questo può essere usato per tracciare

o testare o inviare nuovamente. Quando un lavoro viene completato, verranno inserite anche la data di inizio e la data di fine.

```
[
  {
    "qrtzJobGroup": "PTA0044-PTA0044",
    "qrtzJobName": "PTA0044-168156109957800",
    "status": "HOLD",
    "qrtzDelay": 0,
    "startDate": null,
    "endDate": null,
    "jobName": "PTA0044",
    "userName": "USER1",
    "jobNumber": 9,
    "jobPriority": 5,
    "jobQueue": "queue1",
    "message": {
      "messageQueueName": null,
      "scheduleDate": null,
      "scheduleTime": null,
      "programName": "PTA0044",
      "programParams": {
        "wmind": "B"
      }
    },
    "localDataAreaValue": "",
    "userName": "USER1",
    "jobName": "PTA0044",
    "jobNumber": 9,
    "jobPriority": 5,
    "executionDate": "20181231",
    "jobQueue": "queue1",
    "jobOnHold": true
  },
  {
    "qrtzJobGroup": "PTA0044-PTA0044",
    "qrtzJobName": "PTA0044-166196954877600",
    "status": "COMPLETED",
    "qrtzDelay": 0,
    "startDate": "2022-10-13T22:48:34.025+00:00",
    "endDate": "2022-10-13T22:52:54.475+00:00",
    "jobName": "PTA0044",
    "userName": "USER1",
```

```

    "jobNumber": 9,
    "jobPriority": 5,
    "jobQueue": "queue1",
    "message": {
      "messageQueueName": null,
      "scheduleDate": null,
      "scheduleTime": null,
      "programName": "PTA0044",
      "programParams": {
        "wmind": "B"
      },
      "localDataAreaValue": "",
      "userName": "USER1",
      "jobName": "PTA0044",
      "jobNumber": 9,
      "jobPriority": 5,
      "executionDate": "20181231",
      "jobQueue": "queue1",
      "jobOnHold": true
    }
  }
}
]

```

## Elenco delle risposte alle offerte di lavoro «in sospeso»

La struttura è simile a quella precedente, tranne per il fatto che tutti i lavori restituiti saranno «sospesi».

```

[
  {
    "qrtzJobGroup": "PTA0044-PTA0044",
    "qrtzJobName": "PTA0044-168156109957800",
    "status": "HOLD",
    "qrtzDelay": 0,
    "startDate": null,
    "endDate": null,
    "jobName": "PTA0044",
    "userName": "USER1",
    "jobNumber": 9,
    "jobPriority": 5,
    "jobQueue": "queue1",
    "message": {

```

```
    "messageQueueName": null,  
    "scheduleDate": null,  
    "scheduleTime": null,  
    "programName": "PTA0044",  
    "programParams": {  
      "wmind": "B"  
    },  
    "localDataAreaValue": "",  
    "userName": "USER1",  
    "jobName": "PTA0044",  
    "jobNumber": 9,  
    "jobPriority": 5,  
    "executionDate": "20181231",  
    "jobQueue": "queue1",  
    "jobOnHold": true  
  }  
}  
]
```

## AWS Configurazione Blu Age Runtime (non gestita)

Questa sezione spiega i passaggi per configurare AWS Blu Age Runtime (non gestito) sulla tua AWS infrastruttura.

### Argomenti

- [AWS Prerequisiti di Blu Age Runtime](#)
- [AWS Onboarding di Blu Age Runtime](#)
- [Requisiti di configurazione dell'infrastruttura per AWS Blu Age Runtime \(non gestito\)](#)
- [AWS Implementazione di Blu Age Runtime su Amazon ECS gestita da AWS Fargate](#)
- [AWS Implementazione di Blu Age Runtime su Amazon EC2](#)

## AWS Prerequisiti di Blu Age Runtime

AWS Blu Age Runtime (non gestito) è disponibile in diverse versioni di [rilascio](#). Se avete progetti di modernizzazione in corso, potreste aver bisogno di versioni incrementali del runtime per scopi di implementazione e test. Per definire le tue esigenze, contatta il tuo responsabile delle consegne AWS Blu Age.

Prima di iniziare il processo di onboarding di AWS Blu Age Runtime (non gestito), procedi come segue:

- Assicurati di avere un account AWS
- Assicurati di avere un'applicazione modernizzata rifattorizzata con AWS Blu Age.
- Scegli una AWS regione e un calcolo (Amazon EC2 o Amazon AWS Fargate ECS attivo).
- Scegli la versione di AWS Blu Age Runtime che desideri utilizzare.
- Rivedi [the section called “Requisiti di configurazione dell'infrastruttura”](#) e convalida i componenti aggiuntivi necessari per eseguire AWS Blu Age Runtime (non gestito).

### Note

[Se desideri testare le funzionalità di AWS Blu Age Runtime \(non gestita\), puoi utilizzare l'applicazione demo Planets Demo, che puoi scaricare da -v1.zip. PlanetsDemo](#)

## AWS Onboarding di Blu Age Runtime

Per iniziare, crea un AWS Support caso per richiedere l'onboarding per accedere a AWS Blu Age Runtime. Includi nella richiesta il tuo Account AWS ID, la AWS regione che desideri utilizzare e una scelta di calcolo e una versione di runtime. Se non sei sicuro di quale versione ti serve, contatta il tuo responsabile delle consegne AWS Blu Age.

### AWS Blu Age Runtime (non gestito) su Amazon EC2

Archiviamo gli artefatti AWS Blu Age Runtime (non gestiti) in diversi bucket Amazon S3 per regione e per scelta di elaborazione. Per accedere al bucket per il tuo runtime Regione AWS for AWS Blu Age (non gestito) su Amazon EC2, usa il nome elencato nella tabella seguente.

Regione AWS	Versione Bucket	Costruzione a secchiello
Stati Uniti orientali (Ohio)	aws-bluage-runtime-artifact-s-055777665268-us-east-2	aws-bluage-runtime-artifacts-dev-055777665268-us-east-2
Stati Uniti orientali (Virginia settentrionale)	aws-bluage-runtime-artifact-s-139023371234-us-east-1	aws-bluage-runtime-artifacts-dev-139023371234-us-east-1



Regione AWS	Versione Bucket	Costruzione a secchiello
Stati Uniti occidentali (California settentrionale)	aws-bluage-runtime-artifacts-788454048782-us-west-1	aws-bluage-runtime-artifacts-dev-788454048782-us-west-1
US West (Oregon)	aws-bluage-runtime-artifacts-836771190483-us-ovest-2	aws-bluage-runtime-artifacts-dev-836771190483-us-west-2
Europa (Irlanda)	aws-bluage-runtime-artifacts-925278190477-eu-west-1	aws-bluage-runtime-artifacts-dev-925278190477-eu-west-1
Europa (Parigi)	aws-bluage-runtime-artifacts-673009995881-eu-west-3	aws-bluage-runtime-artifacts-dev-673009995881-eu-west-3
Europa (Francoforte)	aws-bluage-runtime-artifacts-485196800481-eu-central-1	aws-bluage-runtime-artifacts-dev-485196800481-eu-central-1
Sud America (San Paolo)	aws-bluage-runtime-artifacts-737536804457-it-est-1	aws-bluage-runtime-artifacts-dev-737536804457-it-est-1
Asia Pacifico (Tokyo)	aws-bluage-runtime-artifacts-445578176276-ap-nord-est-1	aws-bluage-runtime-artifacts-dev-445578176276-ap-nord-est-1
Asia Pacifico (Sydney)	aws-bluage-runtime-artifacts-726160321909-ap-sudest-2	aws-bluage-runtime-artifacts-dev-726160321909-ap-sudest-2

## AWS Blu Age Runtime (non gestito) su Amazon ECS gestito da Fargate

Archiviamo gli artefatti AWS Blu Age Runtime (non gestiti) in diversi bucket Amazon S3 per regione e per scelta di elaborazione. Per accedere al bucket per il tuo runtime Regione AWS for AWS Blu Age (non gestito) su Amazon ECS gestito da Fargate, usa il nome elencato nella tabella seguente.

Regione AWS	Versione Bucket	Costruzione a secchiello
Stati Uniti orientali (Ohio)	aws-bluage-runtime-fargate-rel-483416914331-us-east-2	aws-bluage-runtime-fargate-dev-483416914331-us-east-2
Stati Uniti orientali (Virginia settentrionale)	aws-bluage-runtime-fargate-rel-308472162679-us-east-1	aws-bluage-runtime-fargate-dev-308472162679-us-east-1
Stati Uniti occidentali (California settentrionale)	aws-bluage-runtime-fargate-rel-343763094578-us-west-1	aws-bluage-runtime-fargate-dev-343763094578-us-west-1
US West (Oregon)	aws-bluage-runtime-fargate-rel-688933007849-us-west-2	aws-bluage-runtime-fargate-dev-688933007849-us-west-2
Europa (Irlanda)	aws-bluage-runtime-fargate-rel-140138033705-eu-west-1	aws-bluage-runtime-fargate-dev-140138033705-eu-west-1
Europa (Parigi)	aws-bluage-runtime-fargate-rel-339712948211-eu-west-3	aws-bluage-runtime-fargate-dev-339712948211-eu-west-3
Europa (Francoforte)	aws-bluage-runtime-fargate-rel-339712918892-eu-central-1	aws-bluage-runtime-fargate-dev-339712918892-eu-central-1
Sud America (San Paolo)	aws-bluage-runtime-fargate-rel-767397998881-it-est-1	aws-bluage-runtime-fargate-dev-767397998881-it-est-1
Asia Pacifico (Tokyo)	aws-bluage-runtime-fargate-rel-891377400849-ap-nord-est-1	aws-bluage-runtime-fargate-dev-891377400849-ap-nord-est-1
Asia Pacifico (Sydney)	aws-bluage-runtime-fargate-rel-533267435478-ap-sudest-2	aws-bluage-runtime-fargate-dev-533267435478-ap-sudest-2

## Utilizzo della riga di comando per elencare il contenuto del bucket

Dopo l'onboarding, puoi elencare il contenuto del bucket eseguendo il seguente comando in un terminale.

```
aws s3 ls bucket-name
```

Sostituiscilo `bucket-name` con il nome del bucket indicato nella tabella Regione AWS precedente.

Questo comando restituisce un elenco di cartelle che corrispondono a diverse versioni del runtime AWS Blu Age Runtime (non gestito), come

```
PRE 3.3.0.1/  
PRE 3.3.0.2/
```

Si consiglia di utilizzare l'ultima versione disponibile. Se ciò non è possibile, utilizzate la versione di runtime convalidata durante la fase di refactoring dell'applicazione. Per elencare i framework disponibili per una versione specifica, esegui il comando seguente:

```
aws s3 ls s3://bucket-name/version/Framework/
```

Sostituiscilo `bucket-name` con il nome del bucket del tuo Regione AWS e `version` con la versione che desideri. Di seguito è riportato un esempio.

```
aws s3 ls s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/  
Framework/
```

Il comando restituirà un elenco di framework, come:

```
2023-06-05 10:26:52 97960225 aws-bluage-runtime-3.4.0.tar.gz  
2023-06-05 10:27:12      45 aws-bluage-runtime-3.4.0.tar.gz.checksumSHA256  
2023-06-05 10:27:14 138497123 aws-bluage-webapps-3.4.0.tar.gz  
2023-06-05 10:27:44      45 aws-bluage-webapps-3.4.0.tar.gz.checksumSHA256
```

## Scarica il framework

Puoi scaricare il framework, ad esempio, per aggiornare la versione Velocity Runtime su un'istanza Amazon EC2 esistente.

```
aws s3 cp s3://bucket-name/version/Framework/ folder-of-your-choice --  
recursive
```

Dove:

`folder-of-your-choice`

percorso della cartella in cui desideri scaricare il framework.

```
Ad esempio: aws s3 cp s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/Framework/ . --recursive
```

Questo comando produce il seguente output:

```
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/Framework/aws-bluage-runtime-3.4.0.tar.gz.checksumSHA256 to ./aws-bluage-runtime-3.4.0.tar.gz.checksumSHA256
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/Framework/aws-bluage-webapps-3.4.0.tar.gz.checksumSHA256 to ./aws-bluage-webapps-3.4.0.tar.gz.checksumSHA256
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/Framework/aws-bluage-webapps-3.4.0.tar.gz to ./aws-bluage-webapps-3.4.0.tar.gz
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/3.4.0/Framework/aws-bluage-runtime-3.4.0.tar.gz to ./aws-bluage-runtime-3.4.0.tar.gz
```

Puoi elencare i file del framework come segue:

```
ls -l
```

Questo comando produce il seguente output:

```
total 230928
-rw-rw-r-- 1 cloudshell-user cloudshell-user 97960225 Jun  5 10:26 aws-bluage-runtime-3.4.0.tar.gz
-rw-rw-r-- 1 cloudshell-user cloudshell-user      45 Jun  5 10:27 aws-bluage-runtime-3.4.0.tar.gz.checksumSHA256
-rw-rw-r-- 1 cloudshell-user cloudshell-user 138497123 Jun  5 10:27 aws-bluage-webapps-3.4.0.tar.gz
-rw-rw-r-- 1 cloudshell-user cloudshell-user      45 Jun  5 10:27 aws-bluage-webapps-3.4.0.tar.gz.checksumSHA256
```

## Requisiti di configurazione dell'infrastruttura per AWS Blu Age Runtime (non gestito)

Questo argomento descrive la configurazione minima dell'infrastruttura richiesta per eseguire AWS Blu Age Runtime (non gestito). Le seguenti procedure descrivono come configurare AWS Blu Age Runtime (non gestito) sul computer prescelto per distribuire un'applicazione modernizzata su Blu Age Runtime. AWS Le risorse che crei devono trovarsi in un Amazon VPC con una sottorete dedicata al dominio dell'applicazione.

## Creazione di un gruppo di sicurezza

1. Apri alla console Amazon VPC all'indirizzo <https://console.aws.amazon.com/vpc/>.
2. Nel riquadro di navigazione a sinistra, in Sicurezza, scegli Gruppi di sicurezza.
3. Nel riquadro centrale, scegli Crea gruppo di sicurezza.
4. Nel campo Nome del gruppo di sicurezza, inserisci **M2BluagePrivateLink-SG**.
5. Nella sezione Regole in entrata, scegliere Aggiungi regola.
6. Per Tipo, scegli HTTPS.
7. Per Source inserisci il tuo VPC CIDR.
8. Nella sezione Regole in uscita, scegli Aggiungi regola.
9. Per Tipo, scegli HTTPS.
10. In Destination (Destinazione), immettere **0.0.0.0/0**.
11. Scegliere Create Security Group (Crea gruppo di sicurezza).

## Crea un endpoint Amazon VPC

1. Apri alla console Amazon VPC all'indirizzo <https://console.aws.amazon.com/vpc/>.
2. Nel riquadro di navigazione a sinistra, sotto Virtual private cloud, scegli Endpoints.
3. Nel riquadro centrale, scegli Crea endpoint.
4. Nella sezione Servizi, inserisci **SQS** il campo di ricerca, quindi seleziona il servizio Amazon SQS corrispondente alla tua regione.
5. Nella sezione VPC, seleziona l'Amazon VPC che hai creato nel passaggio precedente.
6. Nella sezione Subnet, seleziona la sottorete che hai creato per il dominio dell'applicazione.
7. Nella sezione Gruppi di sicurezza, selezionate M2 -SG. BluagePrivateLink
8. Seleziona Crea endpoint.

## Creazione di una policy IAM

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, in Gestione degli accessi, scegli Politiche.
3. Nel riquadro centrale, scegli Crea politica.
4. Nella sezione Editor di policy, scegli l'opzione JSON.

5. Sostituisci tutto il JSON che vedi nell'editor con il seguente JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "sqs:GetQueueUrl",
        "sqs:ReceiveMessage",
        "sqs:SendMessage"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Se hai bisogno di ulteriori dettagli per personalizzare la tua politica, contatta il responsabile delle consegne o l'account manager di AWS Blu Age.

6. Seleziona Avanti.
7. Inserisci un nome per la politica, quindi scegli Crea politica.

### Creazione di un ruolo IAM

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, in Gestione degli accessi, scegli Ruoli.
3. Nel riquadro centrale, scegli Crea ruolo.
4. Nella sezione Caso d'uso, scegli EC2 o Elastic Container Service (quindi Elastic Container Service Task), a seconda della scelta di elaborazione.
5. Seleziona Avanti.
6. Nella casella di ricerca, inserisci il nome della policy che hai creato in precedenza.
7. Seleziona la casella di controllo a sinistra della tua politica.
8. Seleziona Avanti.

9. Immetti un nome del ruolo, quindi scegli Crea ruolo.

## Esecuzione di AWS Blu Age Runtime su Amazon EC2

### Creazione di un'istanza Amazon EC2

1. Aprire la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Scegliere Launch Instance (Avvia istanza).
3. Per Tipo di istanza, scegli uno dei tipi elencati in [the section called “Tipi di istanze Amazon EC2 per AWS Blu Age Runtime \(su Amazon EC2\)”](#).
4. Nella sezione Coppia di chiavi, scegli una coppia di chiavi esistente o creane una nuova.
5. Nella sezione Impostazioni di rete, scegli Seleziona gruppo di sicurezza esistente.
6. Per Gruppi di sicurezza comuni, scegli M2 BluagePrivateLink -SG.
7. Espandi la sezione Dettagli avanzati.
8. Per il profilo dell'istanza IAM, scegli il ruolo IAM che hai creato in precedenza.
9. Scegliere Launch Instance (Avvia istanza).

Quando lo stato dell'istanza Amazon EC2 passa a Running, connettiti all'istanza e installa i seguenti componenti software:

- Java Runtime Environment (JRE) 11.
- Apache Tomcat 9.
- AWS Blu Age Runtime (su Amazon EC2). Installa il runtime AWS Blu Age nella radice della cartella di installazione di Apache Tomcat (alcuni file verranno aggiunti mentre altri verranno sovrascritti).

Se desideri installare webapp aggiuntive, installale in una cartella separata. In questo caso, ripeti il processo di installazione di Apache Tomcat, quindi l'archivio su di esso.

## Esecuzione di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate

Crea un cluster Amazon ECS e un ruolo di attività da utilizzare quando avvii un' AWS Fargate attività in un secondo momento. Per il ruolo dell'attività, includi la policy che hai creato in precedenza. A seconda dello scenario, potrebbe essere necessario allegare politiche IAM aggiuntive al ruolo dell'attività.

## Tipi di istanze Amazon EC2 per AWS Blu Age Runtime (su Amazon EC2)

Di seguito è riportato un elenco dei tipi di istanze Amazon EC2 che puoi utilizzare per AWS Blu Age Runtime (su Amazon EC2).

```
t3.xlarge
t3.small
t3.large
t2.small
t2.large
r6i.xlarge
r6i.large
r6i.4xlarge
r6i.2xlarge
r5b.xlarge
r5b.large
r5b.2xlarge
r3.xlarge
m6i.xlarge
m6i.large
m6i.8xlarge
m6i.4xlarge
m6i.2xlarge
m6i.16xlarge
m5zn.xlarge
m5zn.large
m5zn.3xlarge
m5zn.2xlarge
m5.xlarge
m5.large
m5.8xlarge
m5.4xlarge
m5.2xlarge
m5.16xlarge
m5.12xlarge
c6i.xlarge
c6i.large
c6i.8xlarge
c6i.4xlarge
c6i.2xlarge
c6i.16xlarge
c5.xlarge
c5.large
```



```
c5.9xlarge  
c5.4xlarge  
c5.2xlarge  
c5.18xlarge  
c5.12xlarge
```

## AWS Implementazione di Blu Age Runtime su Amazon ECS gestita da AWS Fargate

Gli argomenti di questa sezione descrivono come configurare AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate, come aggiornare la versione di runtime, come monitorare la distribuzione utilizzando Amazon CloudWatch alarms e come aggiungere dipendenze concesse in licenza.

### Argomenti

- [Configurazione di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate](#)
- [Aggiornamento di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate](#)
- [Amazon CloudWatch Alarms for AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate](#)
- [Configurazione delle dipendenze concesse in licenza in AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate](#)

## Configurazione di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate

Questo argomento spiega come configurare e distribuire l'applicazione di PlanetsDemo esempio utilizzando AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate.

AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate è disponibile per Linux/X86.

### Argomenti

- [Prerequisiti](#)
- [Configurazione](#)
- [Prova l' PlanetsDemo applicazione](#)

### Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Configura il AWS CLI seguendo i passaggi descritti in [Configurazione dell'interfaccia a riga di comando di AWS](#).
- Completa [the section called "AWS Prerequisiti di Blu Age Runtime"](#) e [the section called "AWS Onboarding di Blu Age Runtime"](#)
- Scarica AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate file binari. Per istruzioni, consulta [the section called "AWS Onboarding di Blu Age Runtime"](#).
- Scarica i binari di Apache Tomcat 9.
- [Scarica l'archivio dell'applicazione. PlanetsDemo](#)
- Crea un database Amazon Aurora PostgreSQL per JICS ed esegui la query su di esso. `PlanetsDemo-v1/jics/sql/initJics.sql` Per informazioni su come creare un database Amazon Aurora PostgreSQL, consulta Creazione e [connessione a](#) un cluster Aurora PostgreSQL DB.

## Configurazione

Per configurare l'applicazione di esempio, completa i seguenti passaggi. PlanetsDemo

1. Dopo aver scaricato i file binari di Apache Tomcat, estrai il contenuto e vai alla cartella. `conf` Apri il `catalina.properties` file per modificarlo e sostituisci la riga che inizia `common.loader` con la riga seguente.

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/  
*.jar","${catalina.home}/lib","${catalina.home}/lib/*.jar","${catalina.home}/  
shared","${catalina.home}/shared/*.jar","${catalina.home}/extra","${catalina.home}/  
extra/*.jar"
```

2. Comprimi la cartella Apache Tomcat usando il comando `tar` per creare un archivio ``tar.gz``.
3. Prepara un [Dockerfile](#) per creare l'immagine personalizzata in base ai binari di runtime forniti e ai binari del server Apache Tomcat. Vedi il seguente esempio Dockerfile. L'obiettivo è installare Apache Tomcat 9, seguito da AWS Blu Age Runtime (per Amazon ECS gestito da AWS Fargate) estratto nella radice della directory di installazione di Apache Tomcat 9, e quindi installare l'applicazione modernizzata di esempio denominata. PlanetsDemo

 Note

Il contenuto degli script `install-gapwalk.sh` e `install-app.sh`, utilizzati in questo esempio Dockerfile, è elencato dopo il Dockerfile.

```
FROM --platform=linux/x86_64 amazonlinux:2

RUN mkdir -p /workdir/apps
WORKDIR /workdir
COPY install-gapwalk.sh .
COPY install-app.sh .
RUN chmod +x install-gapwalk.sh
RUN chmod +x install-app.sh

# Install Java and AWS CLI v2-y
RUN yum install sudo java-17-amazon-corretto unzip tar -y
RUN sudo yum remove awscli -y
RUN curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
  "awscliv2.zip"
RUN sudo unzip awscliv2.zip
RUN sudo ./aws/install

# Installation dir
RUN mkdir -p /usr/local/velocity/installation/gapwalk
# Copy PlanetsDemo archive to a dedicated apps dir
COPY PlanetsDemo-v1.zip /workdir/apps/

# Copy resources (tomcat, blu age runtime) to installation dir
COPY tomcat.tar.gz /usr/local/velocity/installation/tomcat.tar.gz
COPY aws-bluage-on-fargate-runtime-3.10.0.15.tar.gz /usr/local/velocity/
  installation/gapwalk/gapwalk-bluage-on-fargate.tar.gz

# run relevant installation scripts
RUN ./install-gapwalk.sh
RUN ./install-app.sh

EXPOSE 8080
EXPOSE 8081
# ...
```

```
# Run Command to start Tomcat server
CMD ["sh", "-c", "sudo /bluage-on-fargate/tomcat.gapwalk/velocity/startup.sh
  $ECS_CONTAINER_METADATA_URI_V4 $AWS_CONTAINER_CREDENTIALS_RELATIVE_URI"]
```

Di seguito sono riportati i contenuti di install-gapwalk.sh.

```
#!/bin/sh

# Vars
TEMP_DIR=/bluage-on-fargate/tomcat.gapwalk/temp

# Install
echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage-on-fargate
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk-bluage-on-fargate.tar.gz
  ${TEMP_DIR}
sudo cp /usr/local/velocity/installation/tomcat.tar.gz ${TEMP_DIR}
# Create velocity dir
mkdir -p /bluage-on-fargate/tomcat.gapwalk/velocity
# Extract tomcat files
tar -xvf ${TEMP_DIR}/tomcat.tar.gz -C ${TEMP_DIR}
# Copy all tomcat files to velocity dir
cp -fr ${TEMP_DIR}/apache-tomcat-9.x.x/* /bluage-on-fargate/tomcat.gapwalk/velocity
# Remove default webapps of Tomcat
rm -f /bluage-on-fargate/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk-bluage-on-fargate.tar.gz -C /bluage-on-fargate/
tomcat.gapwalk
# Remove temporary extraction dir
sudo rm -rf ${TEMP_DIR}
```

Di seguito sono riportati i contenuti di install-app.sh.

```
#!/bin/sh

APP_DIR=/workdir/apps
TOMCAT_GAPWALK_DIR=/bluage-on-fargate/tomcat.gapwalk

unzip ${APP_DIR}/PlanetsDemo-v1.zip -d ${APP_DIR}
cp -r ${APP_DIR}/webapps/* ${TOMCAT_GAPWALK_DIR}/velocity/webapps/
```

```
cp -r ${APP_DIR}/config/* ${TOMCAT_GAPWALK_DIR}/velocity/config/
```

- Fornisci le informazioni di connessione per il database creato come parte dei prerequisiti nel seguente frammento del `application-main.yml` file, che si trova nella cartella `{TOMCAT_GAPWALK_DIR}/config`. Per ulteriori informazioni, consulta [Creazione e connessione a un cluster Aurora PostgreSQL DB](#).

```
datasource:
  jicsDs:
    driver-class-name :
    url:
    username:
    password:
    type :
```

- Crea e trasferisci l'immagine nel tuo repository Amazon ECR. Per istruzioni, consulta [Pushing a Docker image](#) nella Amazon Elastic Container Registry User Guide.
- Apri la console all'indirizzo <https://console.aws.amazon.com/ecs/v2>.
- Nel riquadro di navigazione a sinistra, scegli Definizioni delle attività.
- Per Tipo di avvio, scegli AWS Fargate.
- Seleziona il ruolo dell'attività di cui hai creato [the section called "Requisiti di configurazione dell'infrastruttura"](#).
- Allega l'immagine al contenitore.
- Completa la compilazione del modulo, quindi scegli Crea.
- Nel riquadro di navigazione a sinistra, scegli Cluster, quindi scegli il tuo cluster dall'elenco.
- Nella pagina dei dettagli del cluster, nella scheda Servizi, scegli Crea.
- Seleziona la definizione dell'attività.
- Espandi la sezione Rete e configura il VPC, le sottoreti e il gruppo di sicurezza di cui hai creato come parte. [the section called "Requisiti di configurazione dell'infrastruttura"](#)
- Implementa il tuo servizio Amazon ECS.

Se l'implementazione fallisce, controlla i log. Per trovarli, vai alla pagina delle attività in Amazon ECS gestita da AWS Fargate, quindi scegli la scheda Logs. Se trovi codici di errore che iniziano con una C seguita da un numero, ad esempio CXXXX, prendi nota dei messaggi di errore. Ad esempio, il codice di errore C5102 è un errore comune che indica una configurazione errata dell'infrastruttura. Puoi anche navigare all'interno dell'attività in esecuzione ed eseguire alcuni comandi, in modo simile

a AWS Blu Age Runtime (su Amazon EC2). Per ulteriori informazioni, consulta [Using Amazon ECS Exec per il debug](#) nella Amazon Elastic Container Service Developer Guide.

Per aprire una shell interattiva, esegui il seguente comando dal tuo computer locale.

```
aws ecs execute-command --cluster your_cluster_name --container your_container_name --  
task task_id --interactive --command /bin/sh
```

Prova l' `PlanetsDemo` applicazione

Per verificare lo stato dell' `PlanetsDemo` applicazione distribuita, esegui i seguenti comandi dopo la sostituzione `load-balancer-DNS-name` e `web-binary-name` con i valori corretti per la configurazione. `listener-port`

```
curl http://load-balancer-DNS-name:listener-port/gapwalk-application/
```

Se l'applicazione è in esecuzione, viene visualizzato il seguente messaggio di output: `Jics application is running`.

Quindi, esegui il seguente comando.

```
curl http://load-balancer-DNS-name:listener-port/jac/api/services/rest/jicsservice/
```

Se l'applicazione è in esecuzione, viene visualizzato il seguente messaggio di output: `Jics application is running`.

```
Jics application is running
```

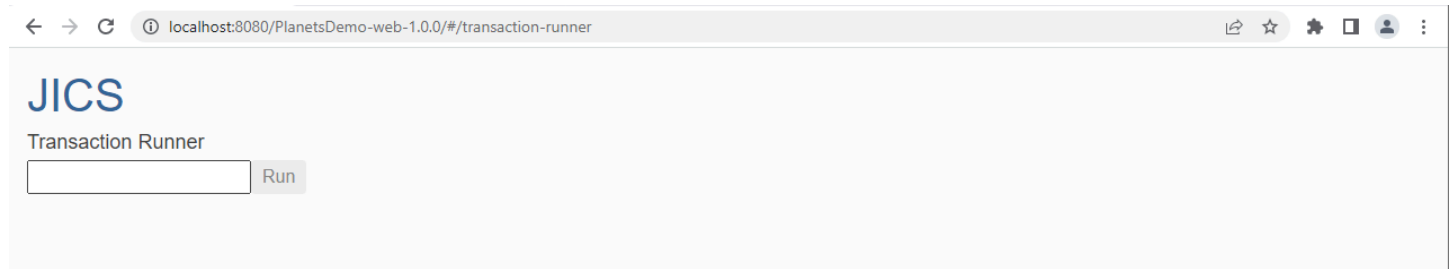
Se hai configurato `Blusam`, puoi aspettarti una risposta vuota quando esegui il comando seguente.

```
curl http://load-balancer-DNS-name:listener-port/bac/api/services/rest/bluesamserver/  
serverIsUp
```

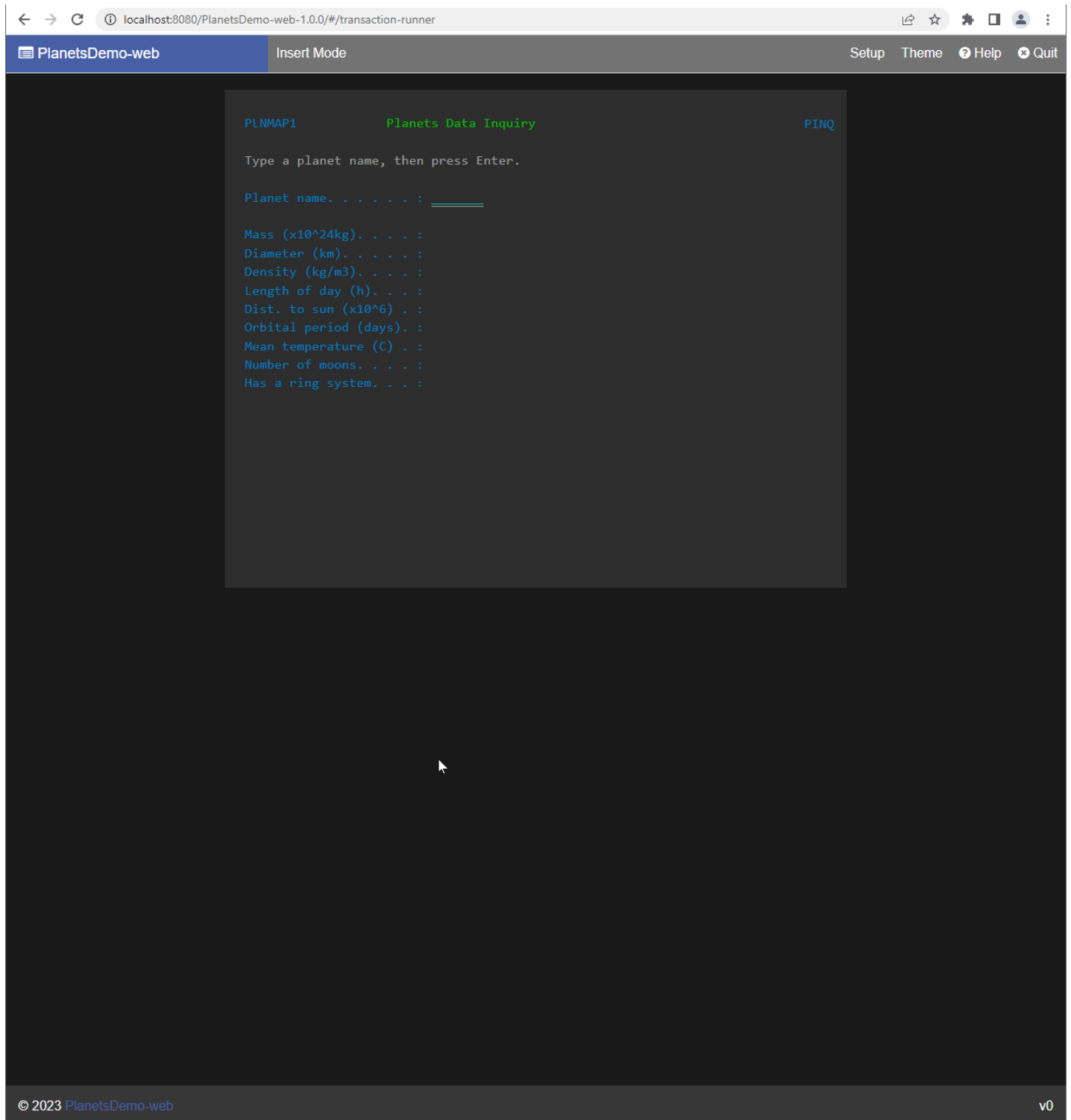
Annotate il nome del file binario web (`PlanetsDemo-web-1.0.0`, se invariato). Per accedere all' `PlanetsDemo` applicazione, utilizzate un URL nel seguente formato.

```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

Dopo l'avvio PlanetsDemo dell'applicazione, viene visualizzata la home page.

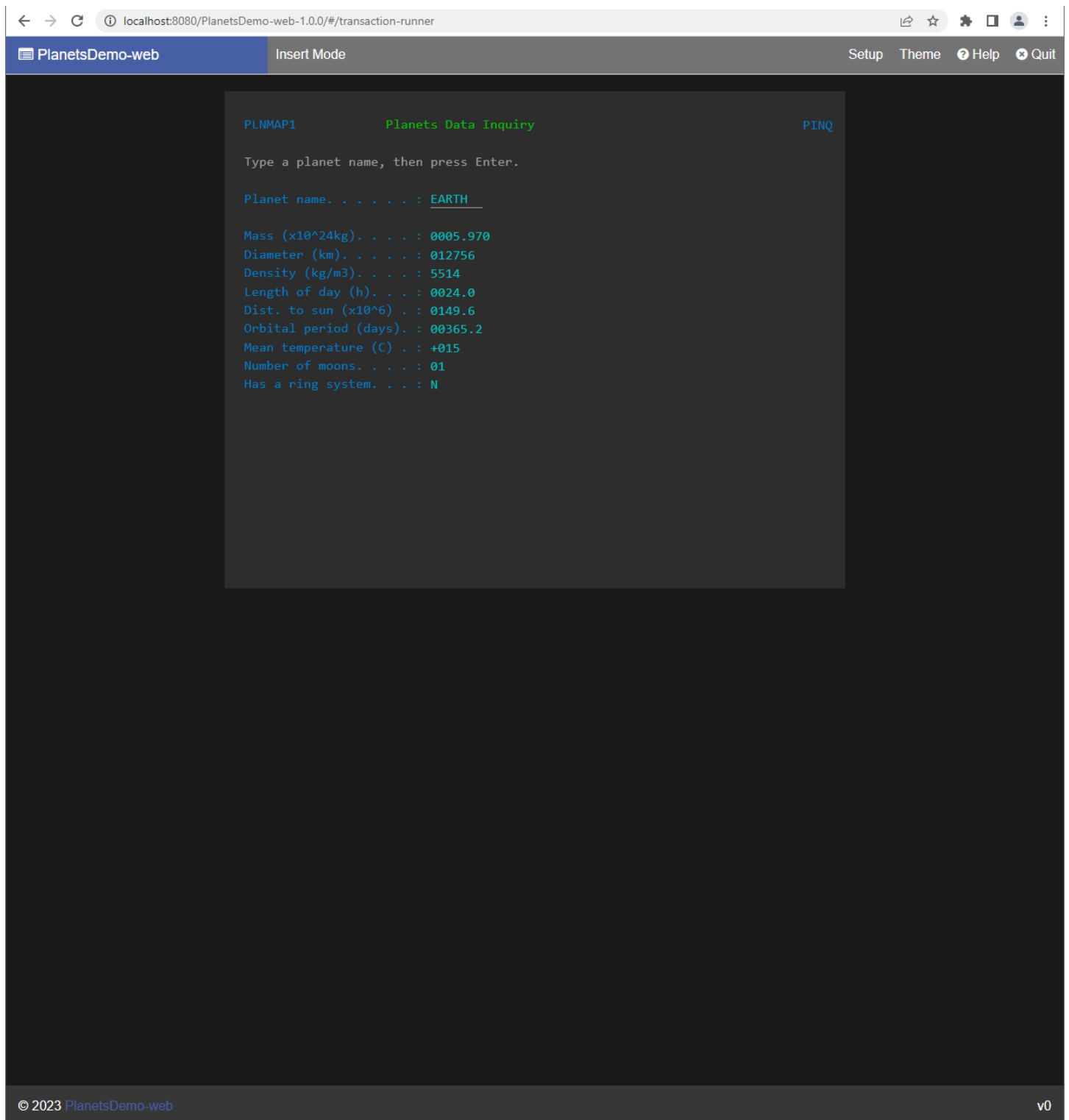


Immettete PINQ nella casella di testo, quindi premete Invio. Viene visualizzata la pagina di richiesta dati.



Ad esempio, inserisci EARTH nel campo del PlanetsDemo nome, quindi premi Invio. Viene visualizzata la pagina relativa al pianeta inserito.





The screenshot shows a web browser window with the address bar at `localhost:8080/PlanetsDemo-web-1.0.0/#/transaction-runner`. The browser title is "PlanetsDemo-web" and the page is in "Insert Mode". The main content is a terminal window titled "Planets Data Inquiry" with a "PINQ" button in the top right corner. The terminal text is as follows:

```
PLNMAP1                               Planets Data Inquiry                               PINQ

Type a planet name, then press Enter.

Planet name. . . . . : EARTH

Mass (x10^24kg). . . . . : 0005.970
Diameter (km). . . . . : 012756
Density (kg/m3). . . . . : 5514
Length of day (h). . . . . : 0024.0
Dist. to sun (x10^6). . . . . : 0149.6
Orbital period (days). . . . . : 00365.2
Mean temperature (C) . . . . . : +015
Number of moons. . . . . : 01
Has a ring system. . . . . : N
```

At the bottom left of the browser window, there is a copyright notice: "© 2023 PlanetsDemo-web". At the bottom right, the version number "v0" is displayed.

## Aggiornamento di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate

Questa guida descrive come aggiornare AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate.

## Argomenti

- [Prerequisiti](#)
- [Aggiorna il velocity runtime](#)

## Prerequisiti

Prima di iniziare, assicurati di soddisfare i seguenti prerequisiti.

- Completa [the section called “AWS Prerequisiti di Blu Age Runtime”](#) e [the section called “AWS Onboarding di Blu Age Runtime”](#)
- Scarica la versione di AWS Blu Age Runtime a cui desideri eseguire l'aggiornamento. Per ulteriori informazioni, consulta [the section called “AWS Onboarding di Blu Age Runtime”](#). Il framework è composto da due file binari: `aws-bluage-runtime-x.x.x.x.tar.gz` e `aws-bluage-webapps-x.x.x.x.tar.gz`.

## Aggiorna il velocity runtime

Completa i seguenti passaggi per aggiornare il velocity runtime.

1. Ricostruisci la tua immagine Docker con la versione AWS Blu Age Runtime desiderata. Per istruzioni, consulta [the section called “Configurazione di AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate”](#).
2. Invia la tua immagine Docker al tuo repository Amazon ECR.
3. Interrompi e riavvia il servizio Amazon ECS.
4. Verifica i log.

Il AWS Blu Age Runtime è stato aggiornato con successo.

## Amazon CloudWatch Alarms for AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate

Per ricevere notifiche più visibili ogni volta che le applicazioni distribuite incontrano eccezioni, configura CloudWatch la ricezione del registro dell'applicazione e aggiungi un allarme per avisarti di possibili errori.

## Configurazione degli allarmi

Con CloudWatch i log, puoi configurare un numero qualsiasi di metriche e allarmi, a seconda dell'applicazione e delle tue esigenze.

In particolare, puoi configurare allarmi proattivi per avvisi di utilizzo direttamente durante la creazione del cluster Amazon ECS, in modo da ricevere notifiche quando si verificano errori. Per evidenziare gli errori nella connessione al sistema di controllo AWS Blu Age, aggiungi una metrica relativa alla stringa «Error C» nei log. È quindi possibile definire un allarme che reagisca a questa metrica.

## Configurazione delle dipendenze concesse in licenza in AWS Blu Age Runtime su Amazon ECS gestito da AWS Fargate

Questo argomento descrive come configurare dipendenze con licenza aggiuntive che puoi utilizzare con AWS Blu Age Runtime on Amazon ECS gestito da AWS Fargate.

### Argomenti

- [Prerequisiti](#)
- [Panoramica](#)

### Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Completa [the section called “AWS Prerequisiti di Blu Age Runtime”](#) e [the section called “AWS Onboarding di Blu Age Runtime”](#)
- Ottieni le seguenti dipendenze dalla loro fonte.

### Oracle database

Fornisci un [driver di database Oracle](#). Ad esempio, ojdbc8-19.8.0.0.jar.

### Connessione IBM MQ

Fornisci un [client IBM MQ](#). Ad esempio, com.ibm.mq.allclient-9.3.0.15.jar.

Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- javax.jms-api-2.0.1.jar
- json-20080701.jar

## file della stampante DDS

Fornisci la libreria di [report Jasper](#). Ad esempio, `jasperreports-6.16.0.jar`, ma una versione più recente potrebbe essere compatibile.

Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- `castor-core-1.4.1.jar`
- `castor-xml-1.4.1.jar`
- `commons-digester-2.1.jar`
- `ecj-3.21.0.jar`
- `itext-2.1.7.js8.jar`
- `javax.inject-1.jar`
- `jcommon-1.0.23.jar`
- `jfreechart-1.0.19.jar`

## Panoramica

Per installare le dipendenze, completa i seguenti passaggi.

1. Copia una delle dipendenze precedenti come richiesto nella cartella di creazione dell'immagine Docker.
2. Se il tuo database JICS o Blusam è ospitato su Oracle, fornisci il driver del database Oracle.  
`your-tomcat-path/extra`
3. Sul tuo Dockerfile, copia queste dipendenze in. `your-tomcat-path/extra`
4. Crea la tua immagine Docker e inviala ad Amazon ECR.
5. Interrompi e riavvia il servizio Amazon ECS.
6. Controllo dei log.

## AWS Implementazione di Blu Age Runtime su Amazon EC2

Gli argomenti di questa sezione descrivono come configurare AWS Blu Age Runtime (non gestito) su Amazon EC2, come aggiornare la versione di runtime, come monitorare la distribuzione utilizzando CloudWatch Amazon alarms e come aggiungere dipendenze concesse in licenza.

## Argomenti

- [Configurazione di AWS Blu Age Runtime \(non gestito\) su Amazon EC2](#)
- [Aggiornamento del AWS Blue Age Runtime su Amazon EC2](#)
- [AWS Blu Age Runtime \(su Amazon EC2\) Amazon Alarms CloudWatch](#)
- [Configurazione delle dipendenze concesse in licenza in AWS Blu Age Runtime su Amazon EC2](#)

## Configurazione di AWS Blu Age Runtime (non gestito) su Amazon EC2

Questo argomento spiega come configurare e distribuire l'applicazione di PlanetsDemo esempio utilizzando AWS Blu Age Runtime (non gestito) su Amazon EC2.

### Argomenti

- [Prerequisiti](#)
- [Configurazione](#)
- [Prova l'applicazione PlanetsDemo](#)

### Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Configura il AWS CLI seguendo i passaggi descritti in [Configurazione dell'interfaccia a riga di comando di AWS](#).
- Completa [the section called "AWS Prerequisiti di Blu Age Runtime"](#) e [the section called "AWS Onboarding di Blu Age Runtime"](#)
- Crea un'istanza Amazon EC2 che contenga l'ultima versione di AWS Blu Age Runtime (su Amazon EC2). Per ulteriori informazioni, consulta [Nozioni di base sulle istanze Amazon EC2 Linux](#).
- Assicurati di poterti connettere correttamente all'istanza Amazon EC2, ad esempio utilizzando SSM.
- Scarica ed estrai AWS Blu Age Runtime (su Amazon EC2) all'indirizzo. *your-tomcat-path*/\* Per istruzioni, consulta [the section called "AWS Onboarding di Blu Age Runtime"](#).
- Scarica l'[archivio PlanetsDemo dell'applicazione](#).
- Decomprimi l'archivio e carica l'applicazione in un bucket Amazon S3 a tua scelta.
- Crea un database Amazon Aurora PostgreSQL per JICS ed esegui la query su di esso. `PlanetsDemo-v1/jics/sql/initJics.sql` Per informazioni su come creare un database

Amazon Aurora PostgreSQL, consulta [Creazione e connessione a un cluster Aurora PostgreSQL DB](#).

## Configurazione

Per configurare l'applicazione di esempio, completa i seguenti passaggi. PlanetsDemo

1. Connettiti alla tua istanza Amazon EC2 e vai alla conf cartella sotto la cartella di installazione di Apache Tomcat 9. Apri il `catalina.properties` file per modificarlo e sostituisci la riga che inizia `common.loader` con la riga seguente.

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/  
*.jar","${catalina.home}/lib","${catalina.home}/lib/*.jar","${catalina.home}/  
shared","${catalina.home}/shared/*.jar","${catalina.home}/extra","${catalina.home}/  
extra/*.jar"
```

2. Accedi alla cartella `<your-tomcat-path>/webapps`.
3. Copia i PlanetsDemo file binari disponibili PlanetsDemo nella cartella `-v1/webapps/` dal bucket Amazon S3 utilizzando il seguente comando.

```
aws s3 cp s3://path-to-demo-app-webapps/ . --recursive
```

### Note

Sostituiscilo `path-to-demo-app-webapps` con l'URI Amazon S3 corretto per il bucket in cui hai precedentemente decompresso l'archivio. PlanetsDemo

4. Copia il contenuto della cartella in. `PlanetsDemo-v1/config/ <your-tomcat-path>/config/`
5. Fornisci le informazioni di connessione per il database che hai creato come parte dei prerequisiti nel seguente frammento del file. `application-main.yml` Per ulteriori informazioni, consulta [Creazione e connessione a un cluster Aurora PostgreSQL DB](#).

```
datasource:  
  jicsDs:  
    driver-class-name :  
    url:  
    username:  
    password:
```

```
type :
```

## 6. Avvia il server Apache Tomcat e verifica i log.

```
your-tomcat-path/startup.sh  
  
tail -f your-tomcat-path/logs/catalina.log
```

Se trovi codici di errore che iniziano con una C seguita da un numero, ad esempio CXXXX, prendi nota dei messaggi di errore. Ad esempio, il codice di errore C5102 è un errore comune che indica una configurazione errata dell'infrastruttura.

## Prova l'applicazione PlanetsDemo

Per verificare lo stato dell' PlanetsDemo applicazione distribuita, esegui i seguenti comandi dopo la sostituzione `load-balancer-DNS-name` e `web-binary-name` con i valori corretti per la configurazione. `listener-port`

```
curl http://load-balancer-DNS-name:listener-port/gapwalk-application/
```

Se l'applicazione è in esecuzione, viene visualizzato il seguente messaggio di output: `Jics application is running`.

Quindi, esegui il seguente comando.

```
curl http://load-balancer-DNS-name:listener-port/jac/api/services/rest/jicsservice/
```

Se l'applicazione è in esecuzione, viene visualizzato il seguente messaggio di output: `Jics application is running`.

```
Jics application is running
```

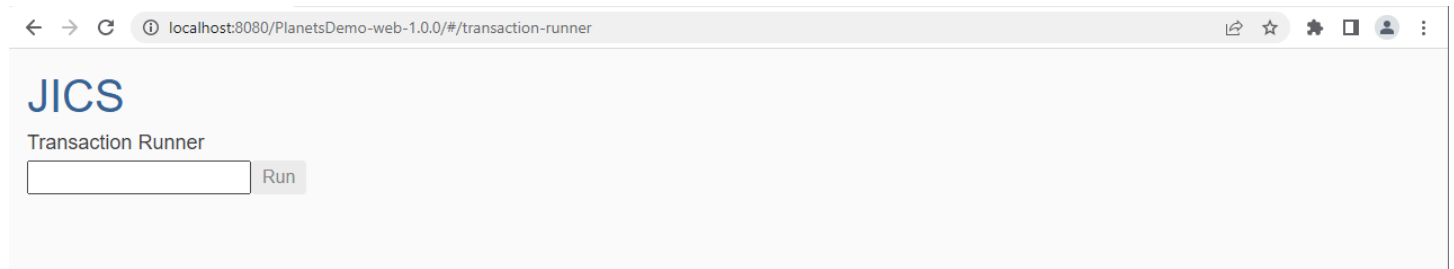
Se hai configurato Blusam, puoi aspettarti una risposta vuota quando esegui il comando seguente.

```
curl http://load-balancer-DNS-name:listener-port/bac/api/services/rest/bluesamserver/  
serverIsUp
```

Annotate il nome del file binario web (PlanetsDemo-web-1.0.0, se invariato). Per accedere all' PlanetsDemo applicazione, utilizzate un URL nel seguente formato.

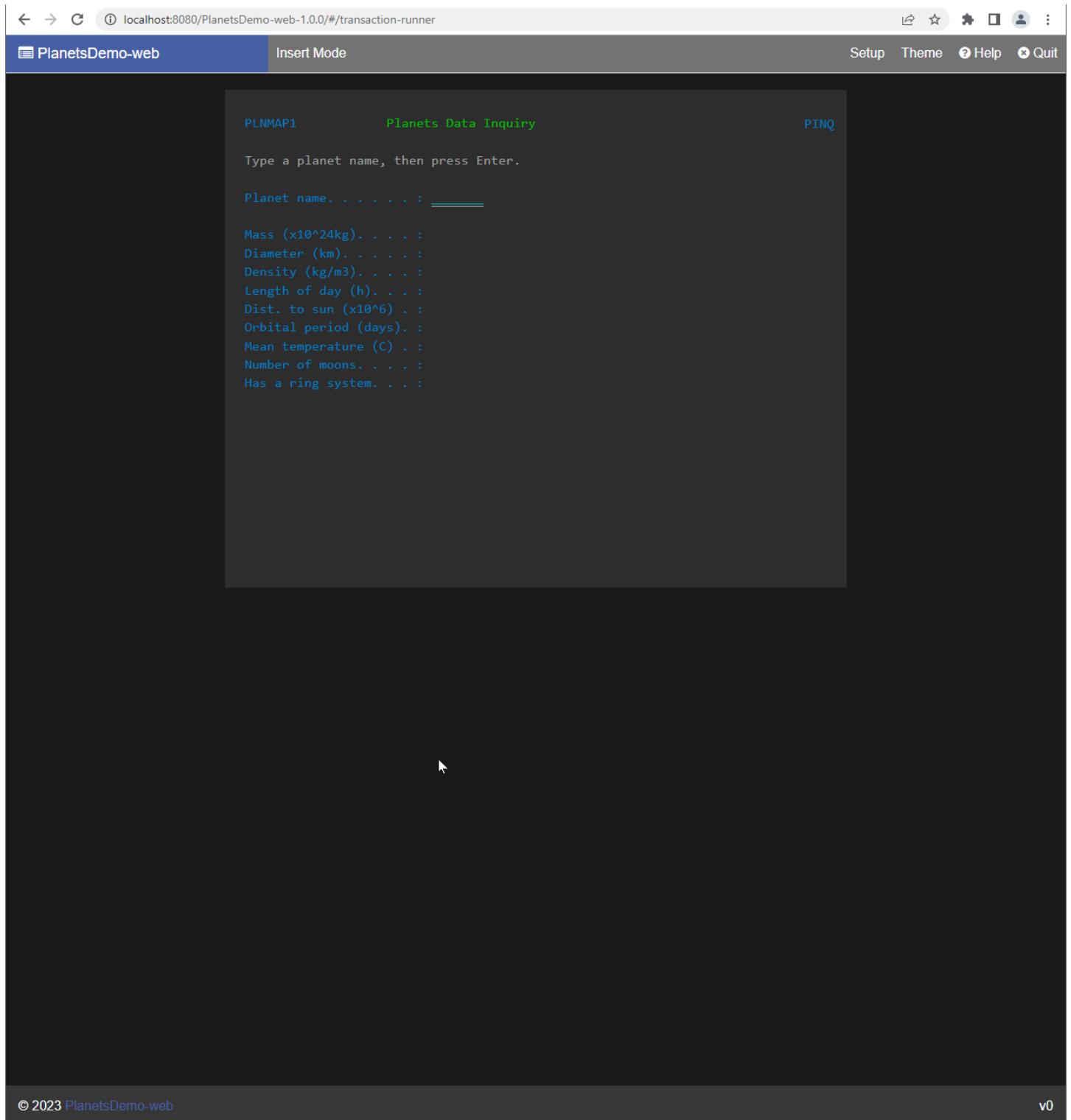
```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

Dopo l'avvio PlanetsDemo dell'applicazione, viene visualizzata la home page.

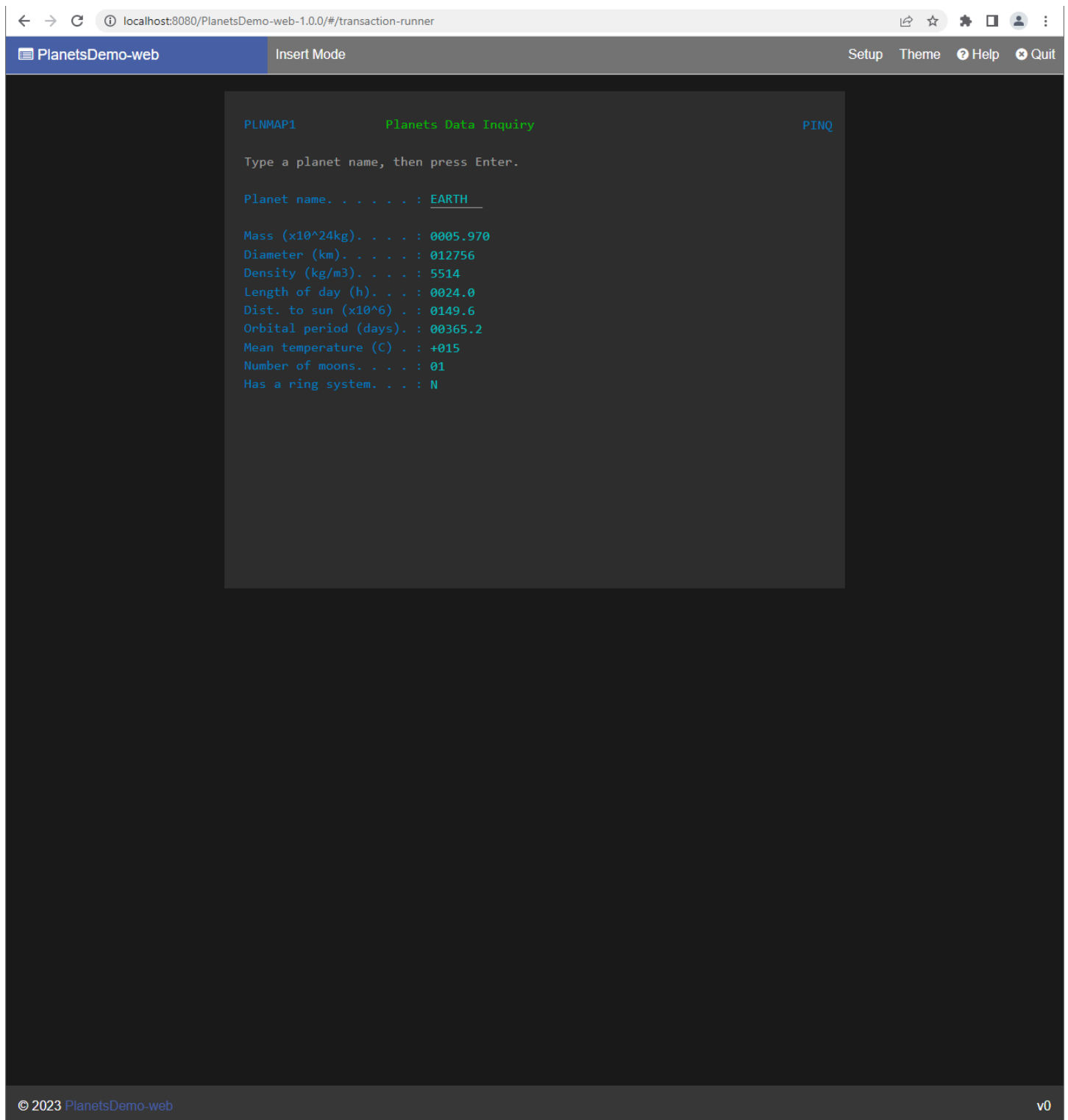


Immettete PINQ nella casella di testo, quindi premete Invio. Viene visualizzata la pagina di richiesta dati.





Ad esempio, inserisci EARTH nel campo del PlanetsDemo nome, quindi premi Invio. Viene visualizzata la pagina relativa al pianeta inserito.



The screenshot shows a web browser window with the address bar at `localhost:8080/PlanetsDemo-web-1.0.0/#/transaction-runner`. The browser title is "PlanetsDemo-web" and the page is in "Insert Mode". The main content is a terminal window titled "Planets Data Inquiry" with a "PINQ" button in the top right corner. The terminal text is as follows:

```
PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

Planet name. . . . . : EARTH

Mass (x10^24kg). . . . . : 0005.970
Diameter (km). . . . . : 012756
Density (kg/m3). . . . . : 5514
Length of day (h). . . . . : 0024.0
Dist. to sun (x10^6). . . . . : 0149.6
Orbital period (days). . . . . : 00365.2
Mean temperature (C) . . . . . : +015
Number of moons. . . . . : 01
Has a ring system. . . . . : N
```

At the bottom of the browser window, there is a footer with the text "© 2023 PlanetsDemo-web" on the left and "v0" on the right.

## Aggiornamento del AWS Blue Age Runtime su Amazon EC2

Questa guida descrive come aggiornare AWS Blu Age Runtime su Amazon EC2.

## Argomenti

- [Prerequisiti](#)
- [Panoramica](#)

## Prerequisiti

Prima di iniziare, assicurati di soddisfare i seguenti prerequisiti.

- Completa [the section called “AWS Prerequisiti di Blu Age Runtime”](#) e [the section called “AWS Onboarding di Blu Age Runtime”](#)
- Assicurati di disporre di un'istanza Amazon EC2 che contenga il runtime AWS Blu Age più recente. Per ulteriori informazioni, consulta [Nozioni di base sulle istanze Amazon EC2 Linux](#).
- Assicurati di poterti connettere correttamente all'istanza Amazon EC2, ad esempio utilizzando SSM.
- Scarica la versione di AWS Blu Age Runtime (su Amazon EC2) a cui desideri eseguire l'aggiornamento. Per ulteriori informazioni, consulta [the section called “AWS Configurazione Blu Age Runtime \(non gestita\)”](#) Il framework è composto da due file binari: `aws-blUAGE-runtime-x.x.x.x.tar.gz` e `aws-blUAGE-webapps-x.x.x.x.tar.gz`.

## Panoramica

Completa i seguenti passaggi per aggiornare il velocity runtime.

1. Connettiti alla tua istanza Amazon EC2 e cambia l'utente in su eseguendo il comando seguente.

```
sudo su
```

È necessario il privilegio Superuser per eseguire i comandi in questo tutorial.

2. Crea due cartelle, una per ogni file binario.
3. Assegna un nome a ogni cartella con lo stesso nome del file binario.
4. Copia ogni file binario nella cartella corrispondente.

**⚠ Warning**

L'estrazione di ogni file binario produce una cartella con lo stesso nome. Pertanto, se estraete entrambi i file binari nella stessa posizione uno dopo l'altro, sovrascriverete il contenuto.

5. Per estrarre i file binari, utilizzate i seguenti comandi. Esegui i comandi in ogni cartella.

```
tar xvf aws-bluage-runtime-x.x.x.x.tar.gz
tar xvf aws-bluage-webapps-x.x.x.x.tar.gz
```

6. Arrestare i servizi Apache Tomcat utilizzando i seguenti comandi.

```
systemctl stop tomcat.service
systemctl stop tomcat-webapps.service
```

7. Sostituisci il contenuto di `<your-tomcat-path>/shared/` con il contenuto di `aws-bluage-runtime-x.x.x.x/velocity/shared/`
8. Sostituisci `<your-tomcat-path>/webapps/gapwalk-application.war` con `aws-bluage-runtime-x.x.x.x/velocity/webapps/gapwalk-application.war`.
9. Sostituisci i file di guerra in `<your-tomcat-path>/webapps/`, vale a dire `bac.war` e `ejac.war`, con gli stessi file `diaws-bluage-webapps-x.x.x.x/velocity/webapps/`.
10. Avvia i servizi Apache Tomcat eseguendo i seguenti comandi.

```
systemctl start tomcat.service
systemctl start tomcat-webapps.service
```

11. Controllo dei log.

Per verificare lo stato dell'applicazione distribuita, esegui i seguenti comandi.

```
curl http://localhost:8080/gapwalk-application/
```

Deve essere visualizzato il seguente messaggio.

```
Jics application is running
```

```
curl http://localhost:8181/jac/api/services/rest/jicsservice/
```

Deve essere visualizzato il seguente messaggio.

```
Jics application is running
```

```
curl http://localhost:8181/bac/api/services/rest/bluesamserver/serverIsUp
```

La risposta dovrebbe essere vuota.

Il runtime AWS Blu Age è stato aggiornato con successo.

## AWS Blu Age Runtime (su Amazon EC2) Amazon Alarms CloudWatch

Per avere notifiche più visibili quando le applicazioni distribuite incontrano eccezioni che impongono all'applicazione un periodo di prova, puoi impostare CloudWatch la ricezione del registro dell'applicazione e aggiungere un allarme per avvisarti di possibili errori.

### Implementazione della registrazione CloudWatch

Per impostazione predefinita, il `application-main.yml` file include un riferimento a un altro file di configurazione di registrazione denominato `logback-cloudwatch.yml`

```
logging:  
  config: classpath:logback-cloudwatch.xml
```

Entrambi i file si trovano nella cartella `config` ed è così che viene configurata CloudWatch la registrazione, come spiegato nelle sezioni seguenti.

### Configurazione della registrazione CloudWatch

Il `logback-cloudwatch.xml` file predefinito ha i seguenti contenuti.

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE configuration>  
<configuration>  
  
  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
```

```

    <encoder>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </encoder>
  </appender>

  <appender name="cloudwatch"
class="com.netfactive.bluage.runtime.cloudwatchlogger.CloudWatchAppender">
    <logGroup>BluAgeRuntimeOnEC2-Logs</logGroup>
    <logStream>%date{yyyy-MM-dd,UTC}.%instanceId.%uuid</logStream>
    <layout>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </layout>
    <appender-ref ref="console" />
  </appender>

  <root level="INFO">
    <appender-ref ref="cloudwatch" />
  </root>
</configuration>

```

Tutto ciò che non è incluso nell'<appender name="cloudwatch"/>elemento è una configurazione di logback standard. In questo file sono presenti due appendici: un'appendice console per inviare i log alla console e un' CloudWatch appendice a cui inviare i log. CloudWatch

L'level attributo nell'root elemento specifica il livello di registrazione dell'intera applicazione.

I valori richiesti all'interno del tag sono: <appender name="cloudwatch"/>

- <logGroup/>: Imposta il nome del gruppo di log. CloudWatch Se il valore non è specificato, il valore predefinito è. BluAgeRuntimeOnEC2-Logs Se il gruppo di log non esiste, verrà creato automaticamente. Questo comportamento può essere modificato tramite la configurazione, descritta di seguito.
- <logStream/>: Imposta il nome di LogStream (all'interno del gruppo di log) in. CloudWatch

Valori opzionali:

- <region/>: sostituisce la regione in cui verrà scritto il flusso di registro. Per impostazione predefinita, i log vanno nella stessa regione dell'istanza EC2.
- <layout/>: Lo schema che verranno utilizzati dai messaggi di registro.

- `<maxbatchsize/>`: Il numero massimo di messaggi di registro a cui inviare CloudWatch per operazione.
- `<maxbatchtimemillis/>`: il tempo in millisecondi per consentire la scrittura CloudWatch dei log.
- `<maxqueuwaittimemillis/>`: Il tempo in millisecondi necessario per provare a inserire le richieste nella coda di registro interna.
- `<internalqueuesize/>`: La dimensione massima della coda interna.
- `<createlogdests/>`: crea un gruppo di log e un flusso di log se non esistono.
- `<initialwaittimemillis/>`: La quantità di tempo in cui desideri che il thread rimanga sospeso all'avvio. Questa attesa iniziale consente un accumulo iniziale di log.
- `<maxeventmessagesize/>`: La dimensione massima di un evento di registro. I log che superano questa dimensione non verranno inviati.
- `<truncateeventmessages/>`: Tronca i messaggi troppo lunghi.
- `<printrejectedevents/>`: abilita l'appendice di emergenza.

## CloudWatch configurazione

Affinché la configurazione precedente invii correttamente i log a CloudWatch, aggiorna il ruolo del profilo dell'istanza IAM di Amazon EC2 per concederle autorizzazioni aggiuntive per il gruppo di log `BluAgeRuntimeOn `EC2-Logs`` e i relativi flussi di log:

- `logs:CreateLogStream`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:DescribeLogGroups`

## Configurazione degli allarmi

Grazie ai CloudWatch log, puoi quindi configurare diverse metriche e allarmi, a seconda dell'applicazione e delle tue esigenze. In particolare, potete impostare allarmi proattivi per avvisi di utilizzo, in modo da essere avvisati in caso di errori che potrebbero mettere l'applicazione in un periodo di prova (e, alla fine, impedirne del tutto il funzionamento). A tal fine, è possibile aggiungere una metrica relativa alla stringa «Error C5001" nei log, che evidenzia gli errori nella connessione

al sistema di controllo Blu Age. AWS È quindi possibile definire un allarme che reagisca a questa metrica.

## Configurazione delle dipendenze concesse in licenza in AWS Blu Age Runtime su Amazon EC2

Questa guida descrive come configurare dipendenze con licenza aggiuntive da utilizzare con AWS Blu Age Runtime su Amazon EC2.

### Argomenti

- [Prerequisiti](#)
- [Panoramica](#)
- [Configura le dipendenze per le webapp JAC e BAC](#)

### Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Completa [the section called “AWS Prerequisiti di Blu Age Runtime”](#) e [the section called “AWS Onboarding di Blu Age Runtime”](#)
- Assicurati di disporre di un'istanza Amazon EC2 contenente l'ultima versione di AWS Blu Age Runtime (su Amazon EC2). Per ulteriori informazioni, consulta [Nozioni di base sulle istanze Amazon EC2 Linux](#).
- Assicurati di poterti connettere correttamente all'istanza Amazon EC2, ad esempio utilizzando SSM.
- Ottieni le seguenti dipendenze dalle loro fonti.

### Oracle database

Fornisci un [driver di database Oracle](#). Abbiamo testato la funzionalità AWS Blu Age Runtime (su Amazon EC2) con la versione ojdbc8-19.8.0.0.jar, ma una versione più recente potrebbe essere compatibile.

### Connessione IBM MQ

Fornisci un [client IBM MQ](#). Abbiamo testato la funzionalità AWS Blu Age Runtime (su Amazon EC2) con la versione com.ibm.mq.allclient-9.3.0.15.jar, ma una versione più recente potrebbe essere compatibile.



Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- javax.jms-api-2.0.1.jar
- json-20080701.jar

file della stampante DDS

Fornisci la libreria di [report Jasper](#). Abbiamo testato la funzionalità AWS Blu Age Runtime (su Amazon EC2) con jasperreports-6.16.0.jar, ma una versione più recente potrebbe essere compatibile.

Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- castor-core-1.4.1.jar
- castor-xml-1.4.1.jar
- commons-digester-2.1.jar
- ecj-3.21.0.jar
- itext-2.1.7.js8.jar
- javax.inject-1.jar
- jcommon-1.0.23.jar
- jfreechart-1.0.19.jar

Panoramica

Per installare le dipendenze, completa i seguenti passaggi.

1. Connettiti alla tua istanza Amazon EC2 e cambia l'utente in su eseguendo il comando seguente.

```
sudo su
```

È necessario il privilegio Superuser per eseguire i comandi in questo tutorial.

2. Accedi alla cartella `<your-tomcat-path>/extra/`.

```
cd <your-tomcat-path>/extra/
```

3. Copia una delle dipendenze precedenti come richiesto in questa cartella.
4. Arresta e avvia tomcat.service eseguendo i seguenti comandi.

```
systemctl stop tomcat.service
```

```
systemctl start tomcat.service
```

5. Controlla lo stato del servizio per assicurarti che sia in esecuzione.

```
systemctl status tomcat.service
```

6. Verifica i log.

### Configura le dipendenze per le webapp JAC e BAC

1. Se il tuo database JICS o Blusam è ospitato su Oracle, devi fornire il driver del database Oracle.  
<your-tomcat-path>/extra
2. Crea la cartella se non è già presente.
3. Arresta e riavvia il server Apache Tomcat.
4. Verifica i log.

## Modifica il codice sorgente con Blu Age Developer IDE

Se utilizzi il motore AWS di runtime AWS Blu Age gestito, puoi utilizzare Blu Age Developer per modificare il codice sorgente generato. Potresti volerlo fare se hai bisogno di aggiornare il codice modernizzato per qualche motivo o se una parte del codice sorgente precedente non può essere modernizzata. Puoi accedere a Blu Age Developer tramite Amazon AppStream 2.0. Questa sezione descrive come configurare Blu Age Developer su AppStream 2.0. Spiega anche come utilizzare Blu Age Developer per aggiornare il codice sorgente, utilizzando l'applicazione di esempio PlanetsDemo.

### Argomenti

- [Tutorial: Configurazione AppStream 2.0 per AWS Blu Age Developer IDE](#)
- [Tutorial: Usa AWS Blu Age Developer su AppStream 2.0](#)

## Tutorial: Configurazione AppStream 2.0 per AWS Blu Age Developer IDE

AWS La modernizzazione del mainframe offre diversi strumenti tramite Amazon AppStream 2.0. AppStream 2.0 è un servizio di streaming di applicazioni completamente gestito e sicuro che

consente di trasmettere applicazioni desktop agli utenti senza riscrivere le applicazioni. AppStream 2.0 offre agli utenti l'accesso immediato alle applicazioni di cui hanno bisogno con un'esperienza utente reattiva e fluida sul dispositivo di loro scelta. L'utilizzo della AppStream versione 2.0 per ospitare strumenti specifici del motore di runtime offre ai team applicativi dei clienti la possibilità di utilizzare gli strumenti direttamente dai propri browser Web, interagendo con i file delle applicazioni archiviati nei bucket o negli archivi Amazon S3. CodeCommit

Per informazioni sul supporto dei browser nella AppStream versione 2.0, consulta [Requisiti di sistema e supporto delle funzionalità \(browser Web\)](#) nella Guida all'amministrazione di Amazon AppStream 2.0. Se riscontri problemi durante l'utilizzo della AppStream versione 2.0, consulta la sezione [Risoluzione dei problemi degli utenti AppStream 2.0](#) nella Guida all'amministrazione di Amazon AppStream 2.0.

Questo documento descrive come configurare AWS Blu Age Developer IDE su una flotta AppStream 2.0.

## Argomenti

- [Prerequisito](#)
- [Fase 1: creazione di un bucket Amazon S3](#)
- [Passaggio 2: allega una policy al bucket S3](#)
- [Passaggio 3: carica i file nel bucket Amazon S3](#)
- [Fase 4: Scarica i modelli AWS CloudFormation](#)
- [Fase 5: Creare la flotta con AWS CloudFormation](#)
- [Fase 6: Accedere a un'istanza](#)
- [Pulizia delle risorse](#)

## Prerequisito

Scarica il [file di archivio](#) che contiene gli artefatti necessari per configurare AWS Blu Age Developer IDE sotto AppStream 2.0.

### Note

Si tratta di un file di grandi dimensioni. In caso di problemi con il timeout dell'operazione, ti consigliamo di utilizzare un'istanza Amazon EC2 per migliorare le prestazioni di upload e download.

## Fase 1: creazione di un bucket Amazon S3

Crea un bucket Amazon S3 nella Regione AWS stessa flotta AppStream 2.0 che creerai. Questo bucket conterrà gli artefatti necessari per completare questo tutorial.

## Passaggio 2: allega una policy al bucket S3

Allega la seguente policy al bucket che crei per questo tutorial. Assicurati di sostituirlo MYBUCKET con il nome effettivo del bucket che hai creato.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowAppStream2.0ToRetrieveObjects",
    "Effect": "Allow",
    "Principal": {
      "Service": "appstream.amazonaws.com"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::MYBUCKET/*"
  }]
}
```

## Passaggio 3: carica i file nel bucket Amazon S3

Decomprimi i file scaricati nel Prerequisito e carica la appstream cartella nel tuo bucket. Il caricamento di questa cartella crea la struttura corretta nel bucket. Per ulteriori informazioni, consulta [Caricamento di oggetti](#) nella Amazon S3 User Guide.

## Fase 4: Scarica i modelli AWS CloudFormation

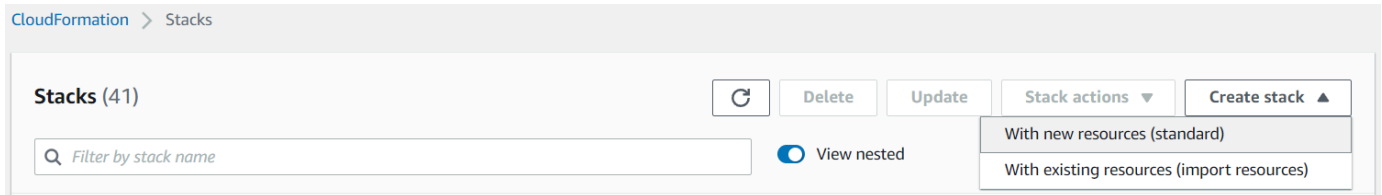
Scarica i seguenti AWS CloudFormation modelli. Questi modelli ti servono per creare e popolare la flotta AppStream 2.0.

- [cfn-m2- .yaml appstream-elastic-fleet-linux](#)
- [cfn-m2- appstream-bluage-dev-tools -linux.yaml](#)
- [cfn-m2 appstream-bluage-shared-linux - .yaml](#)
- [cfn-m2- appstream-chrome-linux .yaml](#)
- [cfn-m2- appstream-eclipse-jee-linux .yaml](#)
- [cfn-m2- appstream-pgadmin-linux .yaml](#)

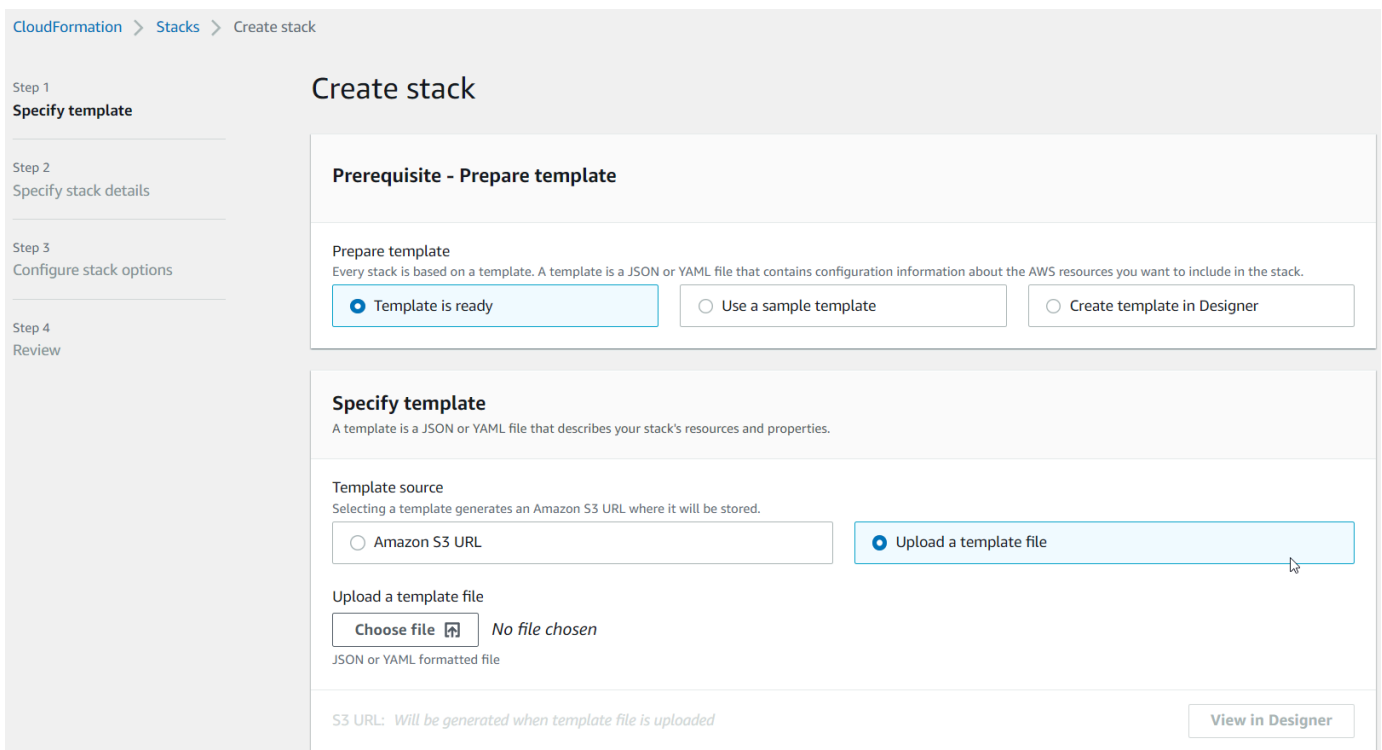
## Fase 5: Creare la flotta con AWS CloudFormation

In questo passaggio, utilizzi il `cfn-m2-appstream-elastic-fleet-linux.yaml` AWS CloudFormation modello per creare una flotta e uno stack AppStream 2.0 per ospitare l'IDE AWS Blu Age Developer. Dopo aver creato la flotta e lo stack, eseguirai gli altri AWS CloudFormation modelli scaricati nel passaggio precedente per installare l'IDE per sviluppatori e gli altri strumenti richiesti.

1. Vai alla console AWS CloudFormation di AWS gestione e scegli Stacks.
2. In Stacks, scegli Crea stack e Con nuove risorse (standard):



3. In Create stack, scegli Il modello è pronto e Carica un file modello:



4. Scegli il file e vai al file `cfn-m2-appstream-elastic-fleet-linux.yaml`. Seleziona Avanti.
5. In Specificare i dettagli dello stack, fornisci le seguenti informazioni:
  - Un nome per lo stack.
  - Il gruppo di sicurezza predefinito e due sottoreti di quel gruppo di sicurezza.

**Note**

Le due sottoreti del gruppo di sicurezza devono trovarsi in zone di disponibilità diverse.

6. Scegli Avanti, quindi scegli nuovamente Avanti.
7. Scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati. , quindi scegli Invia.
8. Dopo aver creato la flotta, crea CloudFormation pile con gli altri modelli scaricati per completare la configurazione delle applicazioni. Assicurati di aggiornare BucketNameogni volta in modo che punti al bucket S3 corretto. Puoi modificarlo BucketNamenella console. CloudFormation In alternativa, puoi modificare direttamente i file modello e aggiornare la S3Bucket proprietà.

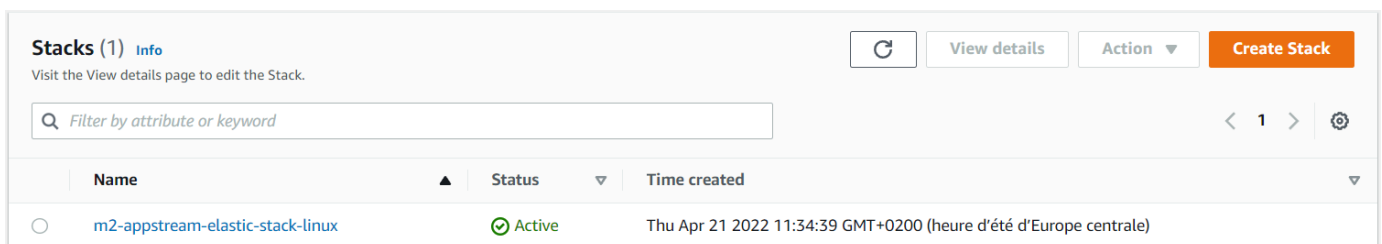
**Note**

I modelli scaricati prevedono di trovare risorse in un bucket S3 con una struttura di cartelle denominata. appstream/bluage/developer-ide/ Il bucket deve appartenere alla Regione AWS stessa flotta che hai creato.

## Fase 6: Accedere a un'istanza

Dopo aver creato e avviato il parco veicoli, puoi creare un link temporaneo per accedere al parco veicoli tramite il client nativo.

1. Passa alla AppStream versione 2.0 AWS Management Console e scegli lo stack creato in precedenza:



The screenshot shows the 'Stacks (1) Info' section in the AWS Management Console. It includes a search bar with the placeholder 'Filter by attribute or keyword', a refresh button, and buttons for 'View details', 'Action', and 'Create Stack'. Below the search bar is a table with the following columns: Name, Status, and Time created. The table contains one entry: 'm2-appstream-elastic-stack-linux' with a status of 'Active' and a creation time of 'Thu Apr 21 2022 11:34:39 GMT+0200 (heure d'été d'Europe centrale)'.

Name	Status	Time created
m2-appstream-elastic-stack-linux	Active	Thu Apr 21 2022 11:34:39 GMT+0200 (heure d'été d'Europe centrale)

2. Nella pagina dei dettagli dello stack, scegli Azione, quindi scegli Crea URL di streaming:

### Create Streaming URL: m2-appstream-elastic-stack-linux ✕

User ID \*

This is the User ID the URL will be associated to.

URL Expiration \*

Set the amount of time the URL will be active before expiration.

30 Minutes ▼

Cancel Get URL

3. In Crea URL di streaming, inserisci un ID utente arbitrario e una data di scadenza dell'URL, quindi scegli Ottieni URL. Ottieni un URL che puoi utilizzare per lo streaming su un browser o sul client nativo. Ti consigliamo di eseguire lo streaming nel client nativo.

## Pulizia delle risorse

Per la procedura di pulizia dello stack e delle flotte creati, consulta [Create an AppStream 2.0 Fleet and Stack](#).

Dopo aver eliminato gli oggetti AppStream 2.0, tu o l'amministratore dell'account potete anche pulire i bucket S3 per le impostazioni dell'applicazione e le cartelle Home.

### Note

La cartella home di un determinato utente è unica per tutte le flotte, quindi potrebbe essere necessario conservarla se nello stesso account sono attivi altri stack AppStream 2.0.

Non è possibile utilizzare la console AppStream 2.0 per eliminare gli utenti. È invece necessario utilizzare l'API del servizio con AWS CLI. Per ulteriori informazioni, consulta [User Pool Administration](#) nella Amazon AppStream 2.0 Administration Guide.

## Tutorial: Usa AWS Blu Age Developer su AppStream 2.0

Questo tutorial mostra come accedere a AWS Blu Age Developer su AppStream 2.0 e utilizzarlo con un'applicazione di esempio in modo da poterne provare le funzionalità. Al termine di questo tutorial, è possibile utilizzare gli stessi passaggi con le proprie applicazioni.

### Argomenti

- [Fase 1: crea un database](#)
- [Fase 2: Accedere all'ambiente](#)
- [Fase 3: Configurare il runtime](#)
- [Passaggio 4: Avvia l'IDE di Eclipse](#)
- [Passaggio 5: configura un progetto Maven](#)
- [Fase 6: Configurare un server Tomcat](#)
- [Passaggio 7: Esegui la distribuzione su Tomcat](#)
- [Fase 8: Creare il database JICS](#)
- [Passaggio 9: Avviare e testare l'applicazione](#)
- [Fase 10: Eseguire il debug dell'applicazione](#)
- [Pulizia delle risorse](#)

### Fase 1: crea un database

In questa fase, utilizzi Amazon RDS per creare un database PostgreSQL gestito che l'applicazione demo utilizza per archiviare le informazioni di configurazione.

1. Apri la console Amazon RDS.
2. Scegli Database > Crea database.
3. Scegli Standard create > PostgreSQL, lascia la versione predefinita, quindi scegli Livello gratuito.
4. Scegli un identificatore di istanza DB.
5. Per Impostazioni delle credenziali, scegli Gestisci le credenziali principali in. AWS Secrets Manager Per ulteriori informazioni, consulta la sezione [Gestione delle password con Amazon RDS e AWS Secrets Manager](#) nella Guida per l'utente di Amazon RDS.
6. Assicurati che il VPC sia lo stesso che usi per l'istanza AppStream 2.0. Puoi chiedere al tuo amministratore questo valore.
7. Per il gruppo di sicurezza VPC, scegli Crea nuovo.



8. Imposta l'accesso pubblico su Sì.
9. Lascia tutti gli altri valori predefiniti. Rivedi questi valori.
10. Scegliere Crea database.

Per rendere il server di database accessibile dalla tua istanza, seleziona il server di database in Amazon RDS. In Connettività e sicurezza, scegli il gruppo di sicurezza VPC per il server del database. Questo gruppo di sicurezza è stato creato in precedenza per te e dovrebbe avere una descrizione simile a quella nella console di gestione Created by RDS. Scegli Azione > Modifica regole in entrata, scegli Aggiungi regola e crea una regola di tipo PostgreSQL. Per l'origine della regola, usa il gruppo di sicurezza predefinito. Puoi iniziare a digitare il nome della fonte nel campo Origine e quindi accettare l'ID suggerito. Infine, scegli Salva regole.

## Fase 2: Accedere all'ambiente

In questo passaggio, accedi all'ambiente di sviluppo AWS Blu Age su AppStream 2.0.

1. Contattate l'amministratore per sapere come accedere correttamente all'istanza AppStream 2.0. Per informazioni generali sui possibili client e configurazioni, consulta la sezione [Metodi e client di accesso AppStream 2.0](#) nella Amazon AppStream 2.0 Administration Guide. Prendi in considerazione l'utilizzo del client nativo per un'esperienza ottimale.
2. Nella AppStream versione 2.0 scegli Desktop.

## Fase 3: Configurare il runtime

In questo passaggio, si configura il runtime AWS Blu Age. È necessario configurare il runtime al primo avvio e di nuovo se si riceve una notifica di aggiornamento del runtime. Questo passaggio popola la .m2 cartella.

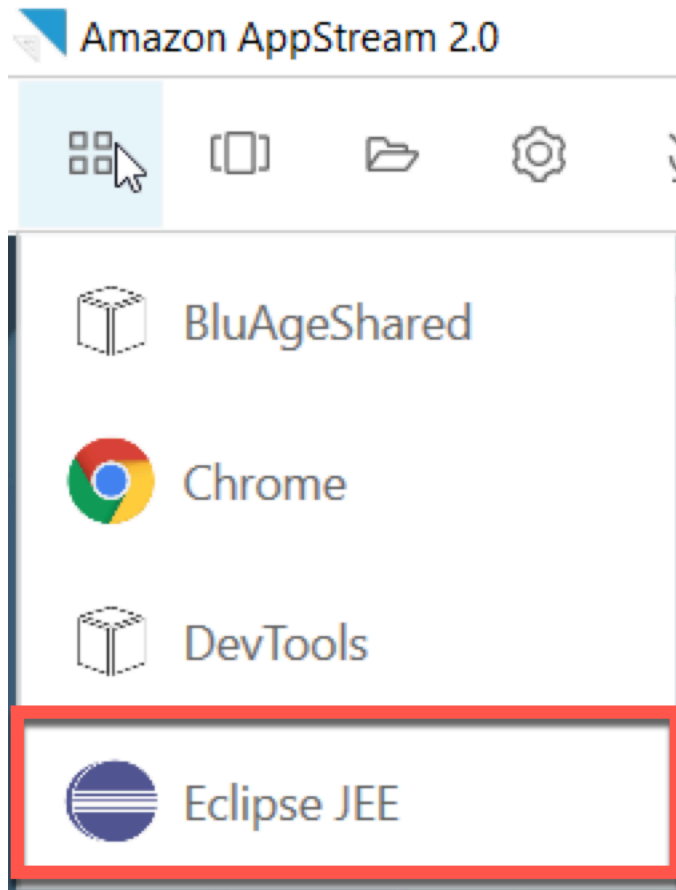
1. Scegli Applicazioni, dalla barra dei menu, quindi scegli Terminale.
2. Immetti il comando seguente:

```
~/_install-velocity-runtime.sh
```

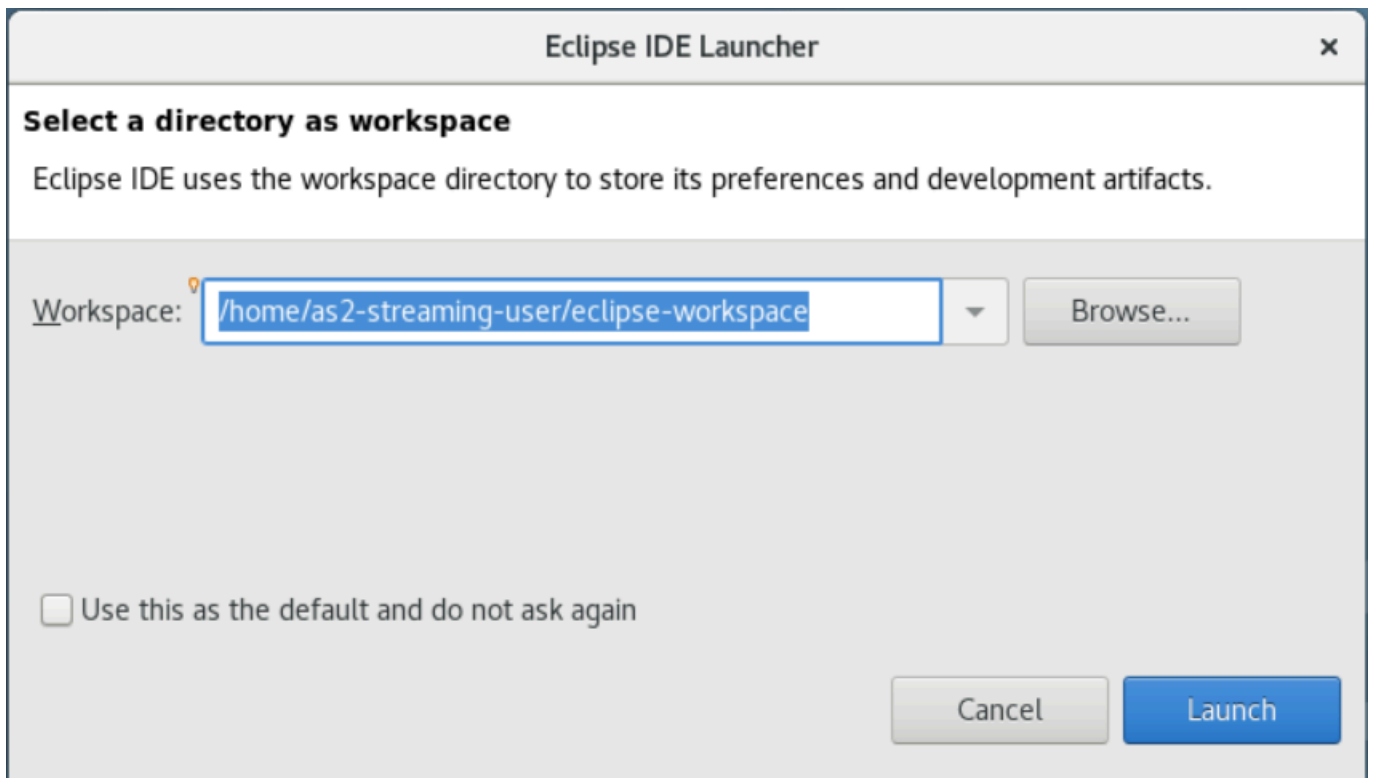
## Passaggio 4: Avvia l'IDE di Eclipse

In questo passaggio, avvii l'IDE di Eclipse e scegli una posizione in cui desideri creare uno spazio di lavoro.

1. Nella AppStream versione 2.0 scegli l'icona Avvia applicazione sulla barra degli strumenti, quindi scegli Eclipse JEE.



2. Quando si apre il programma di avvio, inserisci la posizione in cui desideri creare l'area di lavoro e scegli Avvia.



Facoltativamente, è possibile avviare Eclipse dalla riga di comando, come segue:

```
~/eclipse &
```

## Passaggio 5: configura un progetto Maven

In questo passaggio, si importa un progetto Maven per l'applicazione demo Planets.

1. Carica [PlanetsDemo-pom.zip](#) nella tua cartella Home. A tale scopo, puoi utilizzare la funzione nativa del client «I miei file».
2. Usa lo strumento da riga di unzip comando per estrarre i file.
3. Naviga all'interno della cartella decompressa e apri la radice `pom.xml` del progetto in un editor di testo.
4. Modifica la `gapwalk.version` proprietà in modo che corrisponda al runtime AWS Blu Age installato.

Se non siete sicuri della versione installata, eseguite il seguente comando in un terminale:

```
cat ~/runtime-version.txt
```

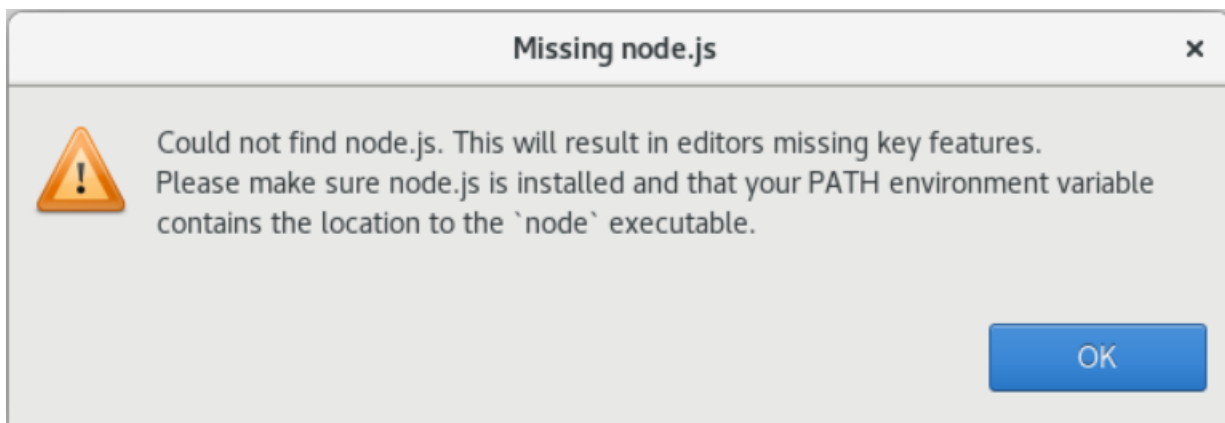
Questo comando stampa la versione di runtime attualmente disponibile, `3.1.0-b3257-dev` ad esempio.

#### Note

Non includere il `-dev` suffisso in `gapwalk.version`. Ad esempio, un valore valido sarebbe `<gapwalk.version>3.1.0-b3257</gapwalk.version>`.

5. In Eclipse, scegli File, quindi Importa. Nella finestra di dialogo Importa, espandi Maven e scegli Progetti Maven esistenti. Seleziona Avanti.
6. In Importa progetti Maven, fornisci la posizione dei file estratti e scegli Fine.

Puoi tranquillamente ignorare il seguente popup. Maven scarica una copia locale di `node.js` per creare la parte Angular (`*-web`) del progetto:



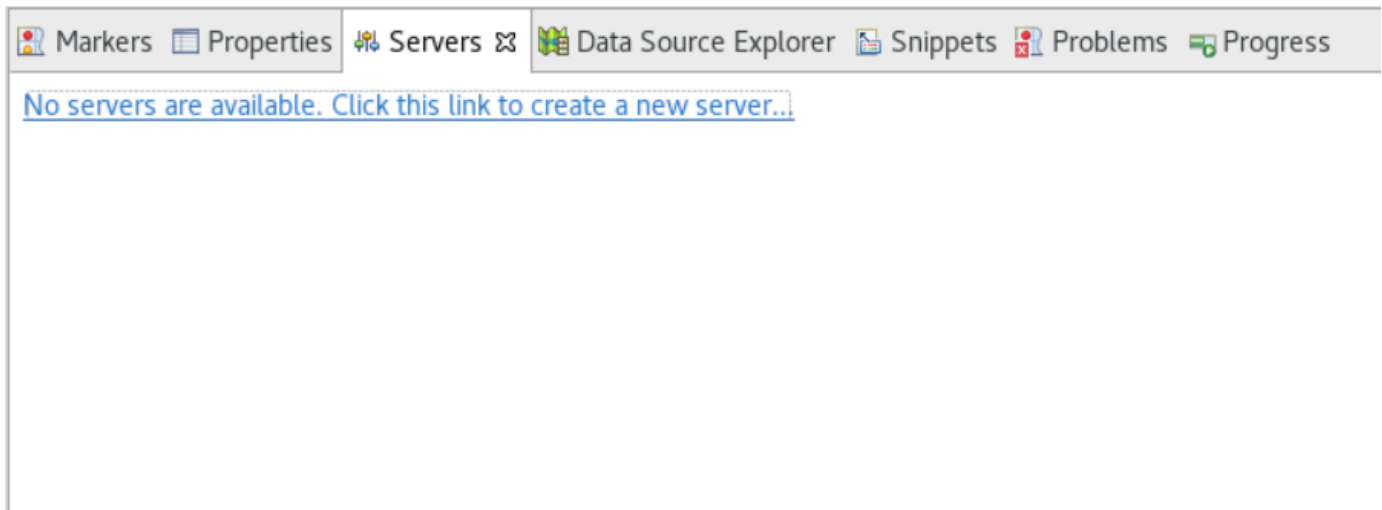
Attendi fino alla fine della compilazione. Puoi seguire la build nella vista Progress.

7. In Eclipse, seleziona il progetto e scegli Esegui come. Quindi scegli Maven install. Una volta completata l'installazione di Maven, crea il file seguente. `war PlanetsDemoPom/PlanetsDemo-web/target/PlanetsDemo-web-1.0.0.war`

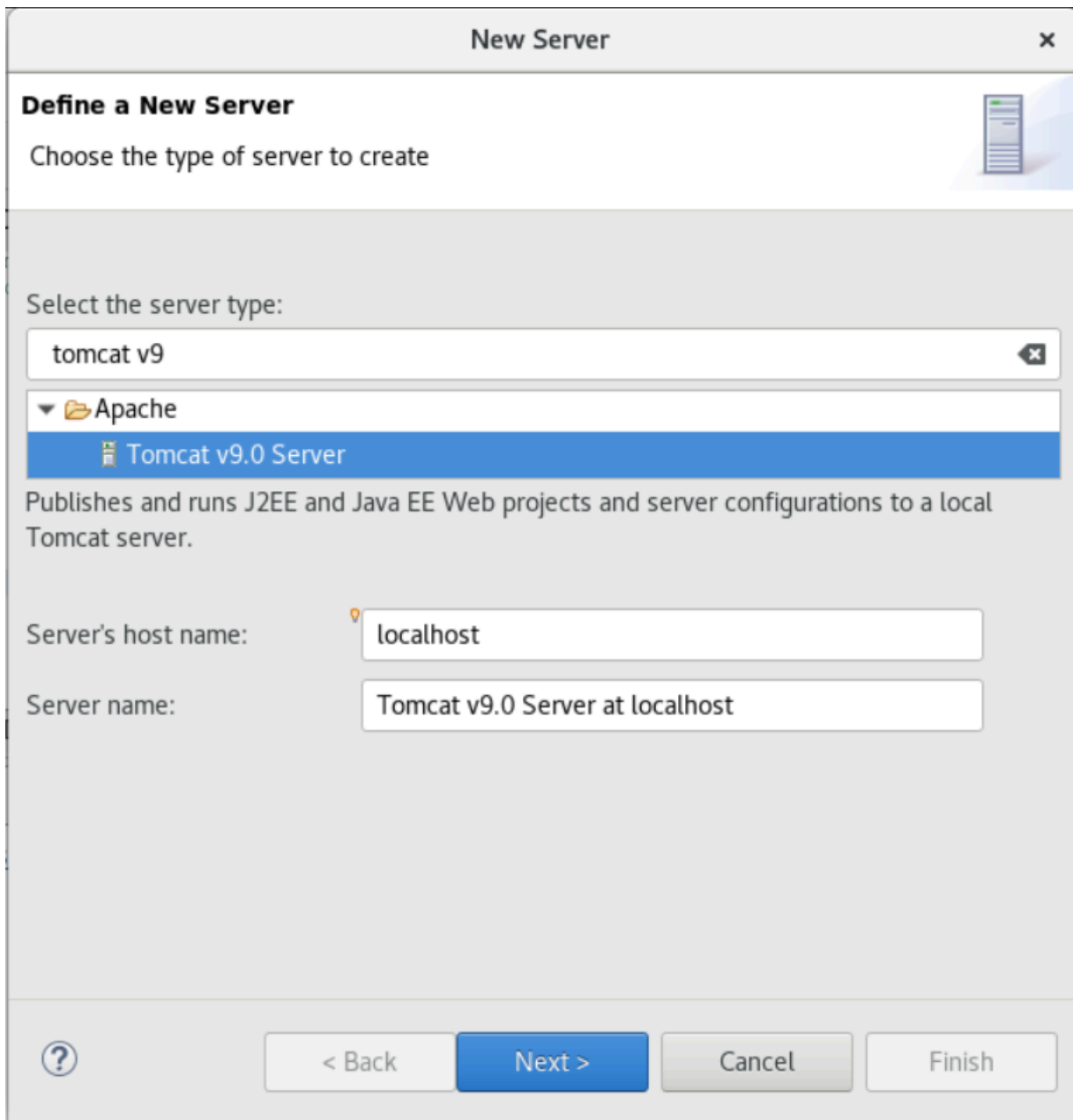
## Fase 6: Configurare un server Tomcat

In questo passaggio, si configura un server Tomcat in cui distribuire e avviare l'applicazione compilata.

1. In Eclipse, scegli Finestra > Mostra vista > Server per mostrare la vista dei server:

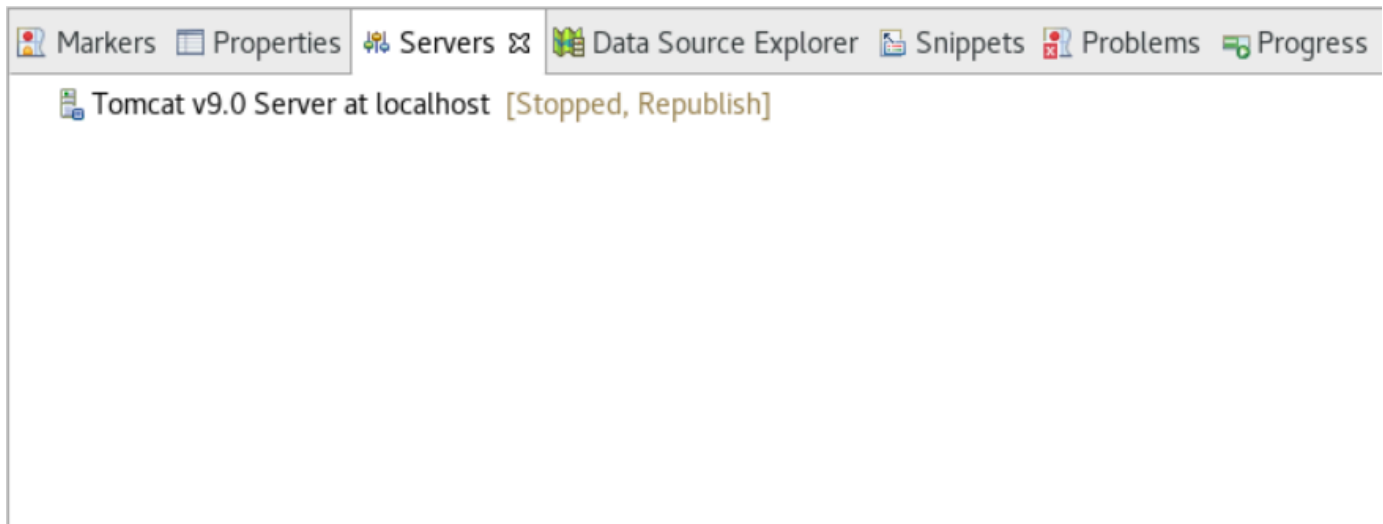


2. Scegli Non ci sono server disponibili. Fai clic su questo link per creare un nuovo server...  
Viene visualizzata la procedura guidata New Server. Nel campo Seleziona il tipo di server della procedura guidata, inserisci tomcat v9 e scegli Tomcat v9.0 Server. Quindi scegli Successivo.



3. Scegli Sfoglia e scegli la cartella tomcat nella radice della cartella Home. Lascia il valore predefinito di JRE e scegli Fine.

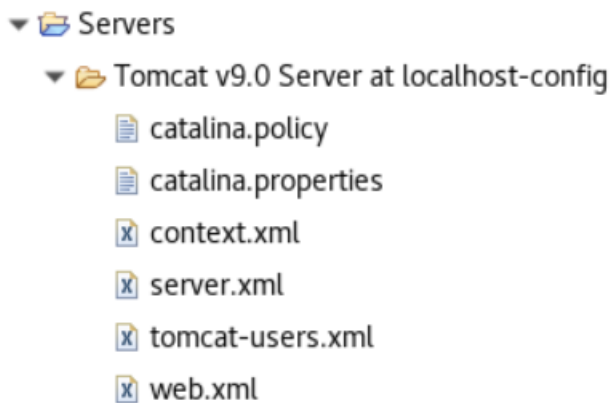
Un progetto Servers viene creato nell'area di lavoro e un server Tomcat v9.0 è ora disponibile nella vista Server. Qui verrà distribuita e avviata l'applicazione compilata:



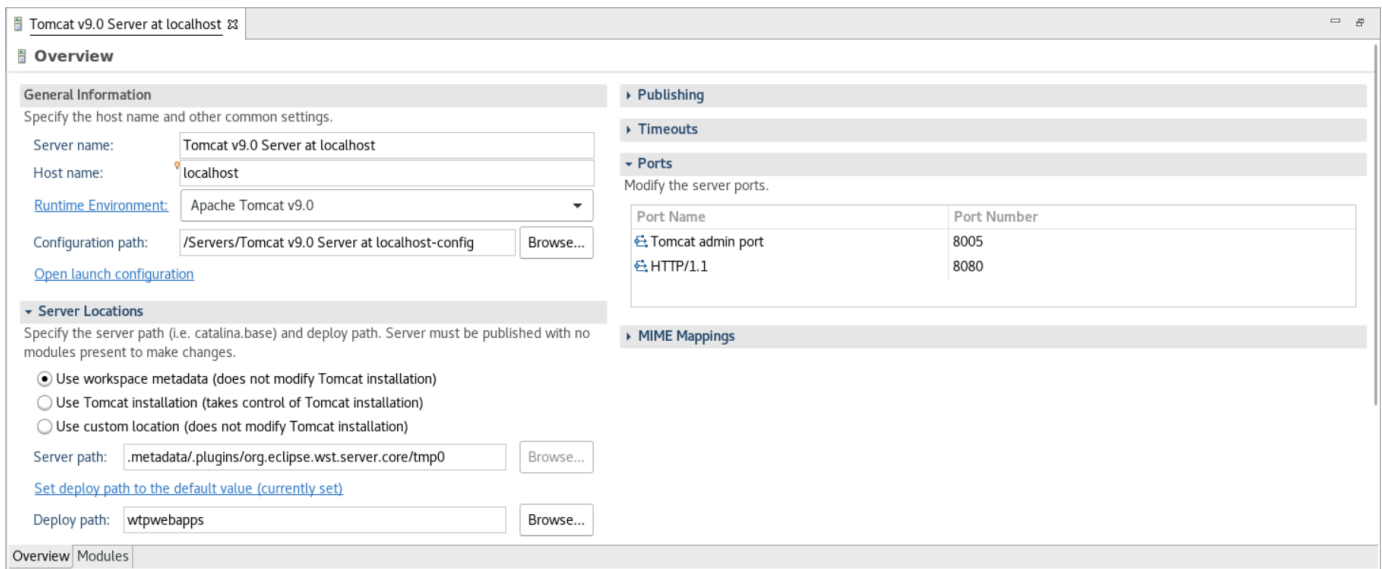
## Passaggio 7: Esegui la distribuzione su Tomcat

In questo passaggio, distribuisce l'applicazione demo Planets sul server Tomcat in modo da poter eseguire l'applicazione.

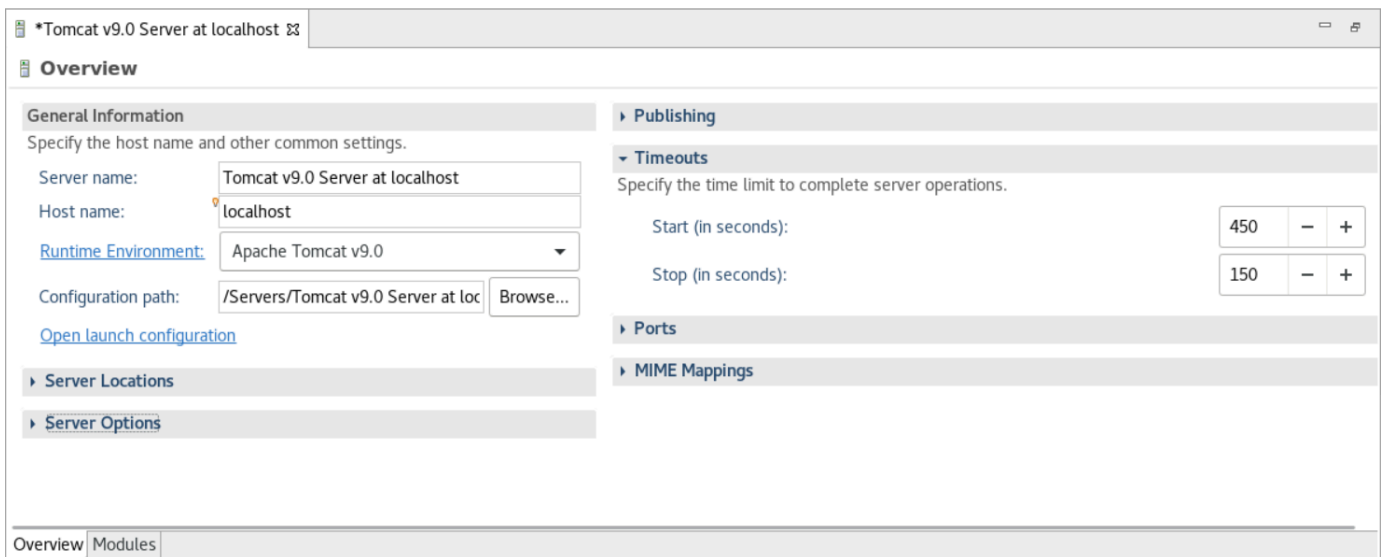
1. Seleziona il `PlanetsDemo-web` file e scegli **Esegui come > Maven install**. Seleziona `PlanetsDemo-web` di nuovo e scegli **Aggiorna** per assicurarti che il frontend compilato con npm sia correttamente compilato in `.war` e notato da Eclipse.
2. Carica [PlanetsDemo-runtime.zip](#) sull'istanza e decomprimi il file in una posizione accessibile. Ciò garantisce che l'applicazione demo possa accedere alle cartelle e ai file di configurazione richiesti.
3. Copia il contenuto di `PlanetsDemo-runtime/tomcat-config` nella `Servers/Tomcat v9.0...` sottocartella che hai creato per il tuo server Tomcat:



4. Aprire la voce relativa al tomcat v9.0 server nella vista Server. Viene visualizzato l'editor delle proprietà del server:

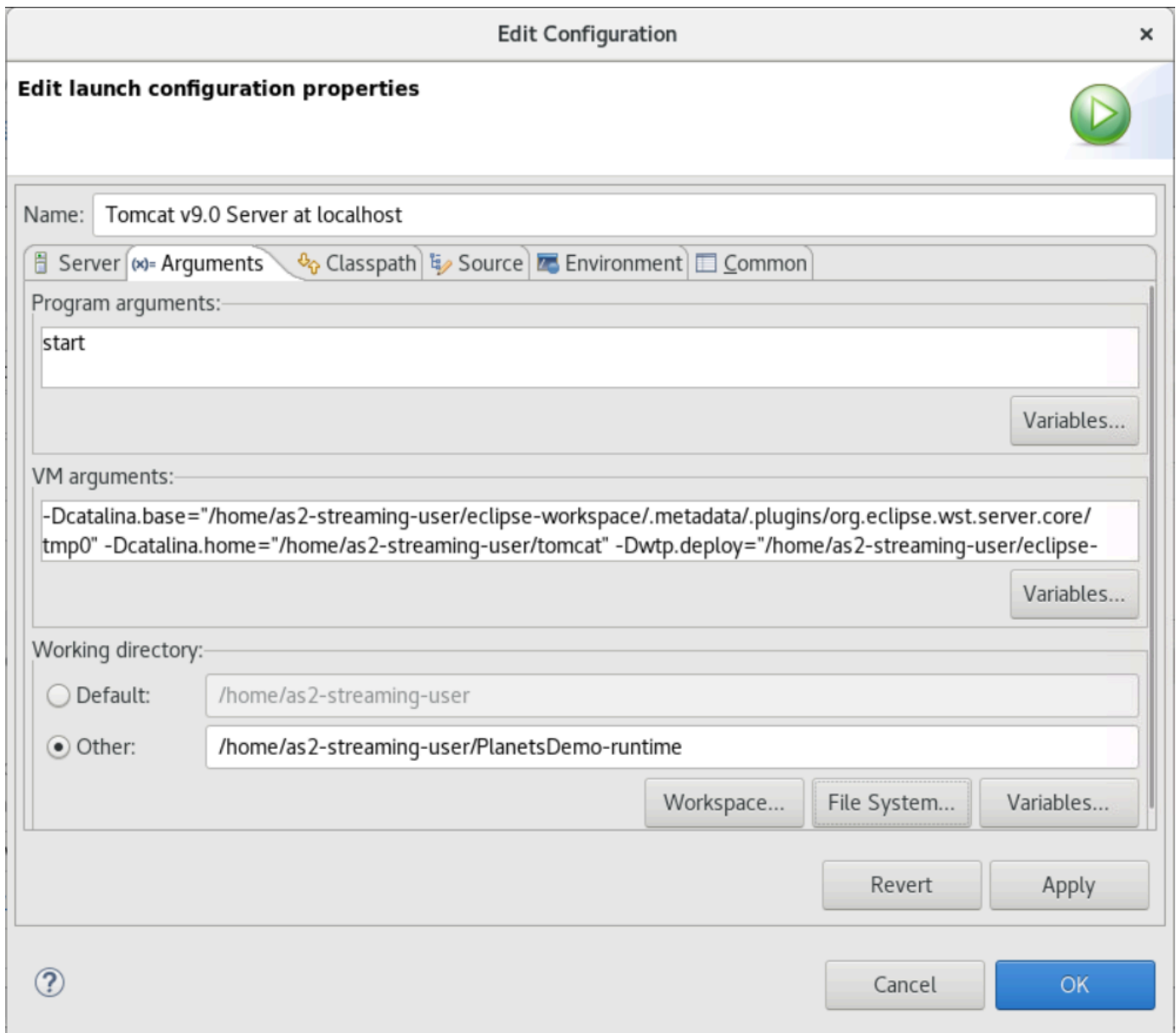


5. Nella scheda Panoramica, aumenta i valori dei timeout a 450 secondi per Start e 150 secondi per Stop, come illustrato di seguito:



6. Scegli Open launch configuration. Viene visualizzata una procedura guidata. Nella procedura guidata, accedete alla cartella Arguments e, per Working directory, scegliete Altro. Scegliete File System e accedete alla PlanetsDemo-runtime cartella decompressa in precedenza. Questa cartella dovrebbe contenere una sottocartella diretta chiamata config.



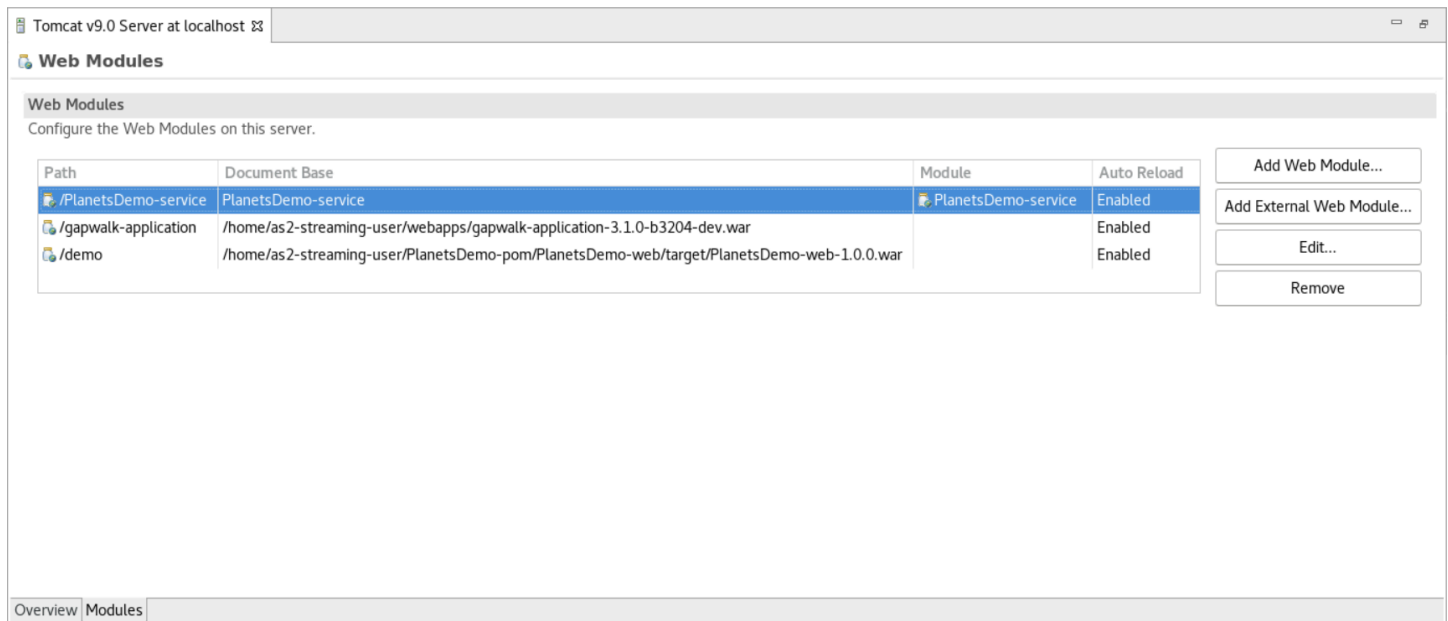


7. Scegliete la scheda Moduli dell'editor delle proprietà del server e apportate le seguenti modifiche:

- Scegli Aggiungi modulo Web e aggiungi PlanetsDemo-service.
- Scegli Aggiungi modulo Web esterno. Viene visualizzata la finestra di dialogo Aggiungi modulo Web. Apporta le seguenti modifiche:
  - In Document Base, scegliete Sfoglia e accedete a `~/webapps/gapwalk-application...war`
  - In Path, inserisci `/gapwalk-application`.
- Scegli OK.
- Scegliete nuovamente Aggiungi modulo Web esterno e apportate le seguenti modifiche:

- Per Document base, inserisci il percorso del frontend .war (in) PlanetsDemo-web/target
- Per Path, immettete /demo
- Seleziona OK
- Salva le modifiche all'editor (Ctrl + S).

Il contenuto dell'editor dovrebbe ora essere simile al seguente.



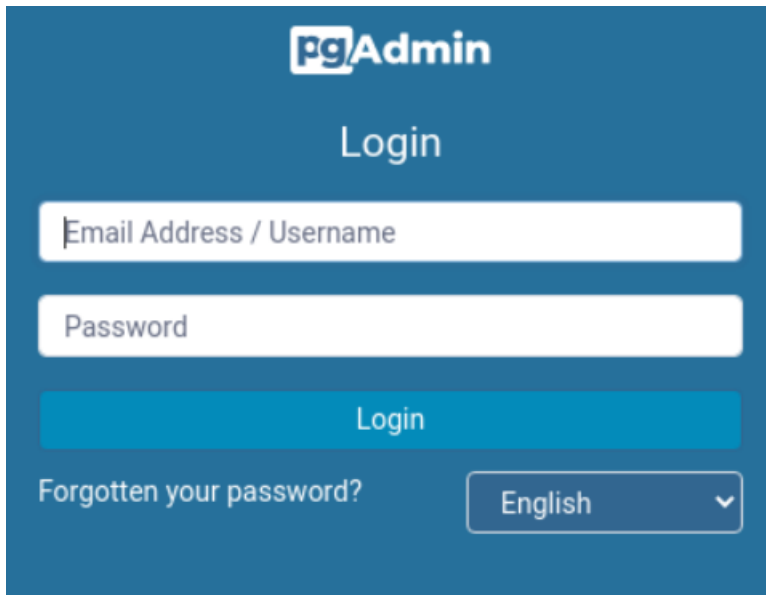
## Fase 8: Creare il database JICS

In questo passaggio, ti connetti al database in [Fase 1: crea un database](#) cui hai creato.

1. Dall'istanza AppStream 2.0, esegui il seguente comando in un terminale per avviare pgAdmin:

```
./pgadmin-start.sh
```

2. Scegli un indirizzo email e una password come identificatori di accesso. Prendi nota dell'URL fornito (in genere `http://127.0.0.1:5050`). Avvia Google Chrome nell'istanza, copia e incolla l'URL nel browser e accedi con i tuoi identificatori.



3. Dopo aver effettuato l'accesso, scegli Aggiungi nuovo server e inserisci le informazioni di connessione al database creato in precedenza come segue.

**Register - Server**

General **Connection** SSL SSH Tunnel Advanced

Host name/address: xxx.yyy.zzz.rds.amazonaws.com

Port: 5432

Maintenance database: postgres

Username: postgres

Kerberos authentication?

Password: .....

Save password?

Role:

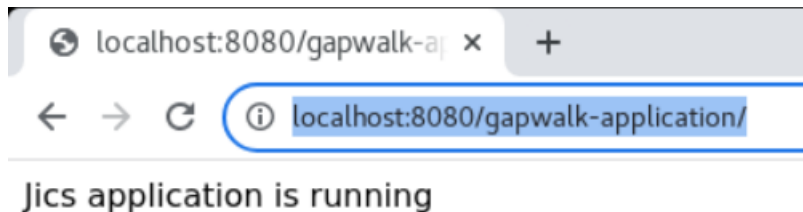
Service:

- Quando ti connetti al server del database, usa Oggetto > Crea > Database e crea un nuovo database denominato jics.
- Modifica le informazioni di connessione al database utilizzate dall'app demo. Queste informazioni sono definite in `PlanetsDemo-runtime/config/application-main.yml`. Cerca la `jicsDs` voce. Per recuperare i valori per `username` e `password`, nella console Amazon RDS, accedi al database. Nella scheda Configuration (Configurazione), in Master Credentials ARN (ARN delle credenziali principali), scegli Manage in Secrets Manager (gestisci in secrets manager). Quindi, nella console di Secrets Manager, nel campo segreto, scegli Recupera valore segreto.

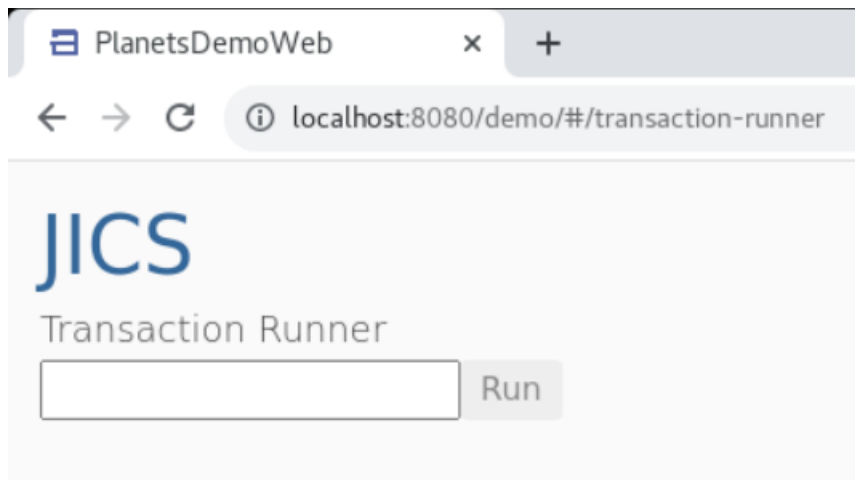
## Passaggio 9: Avviare e testare l'applicazione

In questo passaggio, si avvia il server Tomcat e l'applicazione demo in modo da poterla testare.

1. Per avviare il server Tomcat e le applicazioni precedentemente distribuite, seleziona la voce relativa al server nella vista Server e scegli Avvia. Viene visualizzata una console che mostra i registri di avvio.
2. Controlla lo stato del server nella vista Server o attendi il messaggio di avvio del server tra [xxx] millisecondi nella console. Dopo l'avvio del server, verifica che gapwalk-application sia distribuita correttamente. Per fare ciò, accedi all'URL `http://localhost:8080/gapwalk-application` in un browser Google Chrome. Dovresti vedere quanto segue.



3. Accedi al frontend dell'applicazione distribuita da Google Chrome all'indirizzo `http://localhost:8080/demo`. Dovrebbe apparire la seguente pagina Transaction Launcher.



4. Per avviare la transazione dell'applicazione, inserisci PINQ nel campo di immissione e scegli Esegui (o premi Invio).

Dovrebbe apparire la schermata dell'app demo.

```
PlanetsDemo-web  Insert Mode  Setup Theme Help Quit

PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

Planet name. . . . . _____

Mass (x10^24kg). . . . . :
Diameter (km). . . . . :
Density (kg/m3). . . . . :
Length of day (h). . . . :
Dist. to sun (x10^6) . . :
Orbital period (days). . :
Mean temperature (C) . . :
Number of moons. . . . . :
Has a ring system. . . . :
```

5. Digita il nome di un pianeta nel campo corrispondente e premi Invio.

```
PlanetsDemo-web  Insert Mode  Setup Theme Help Quit

PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

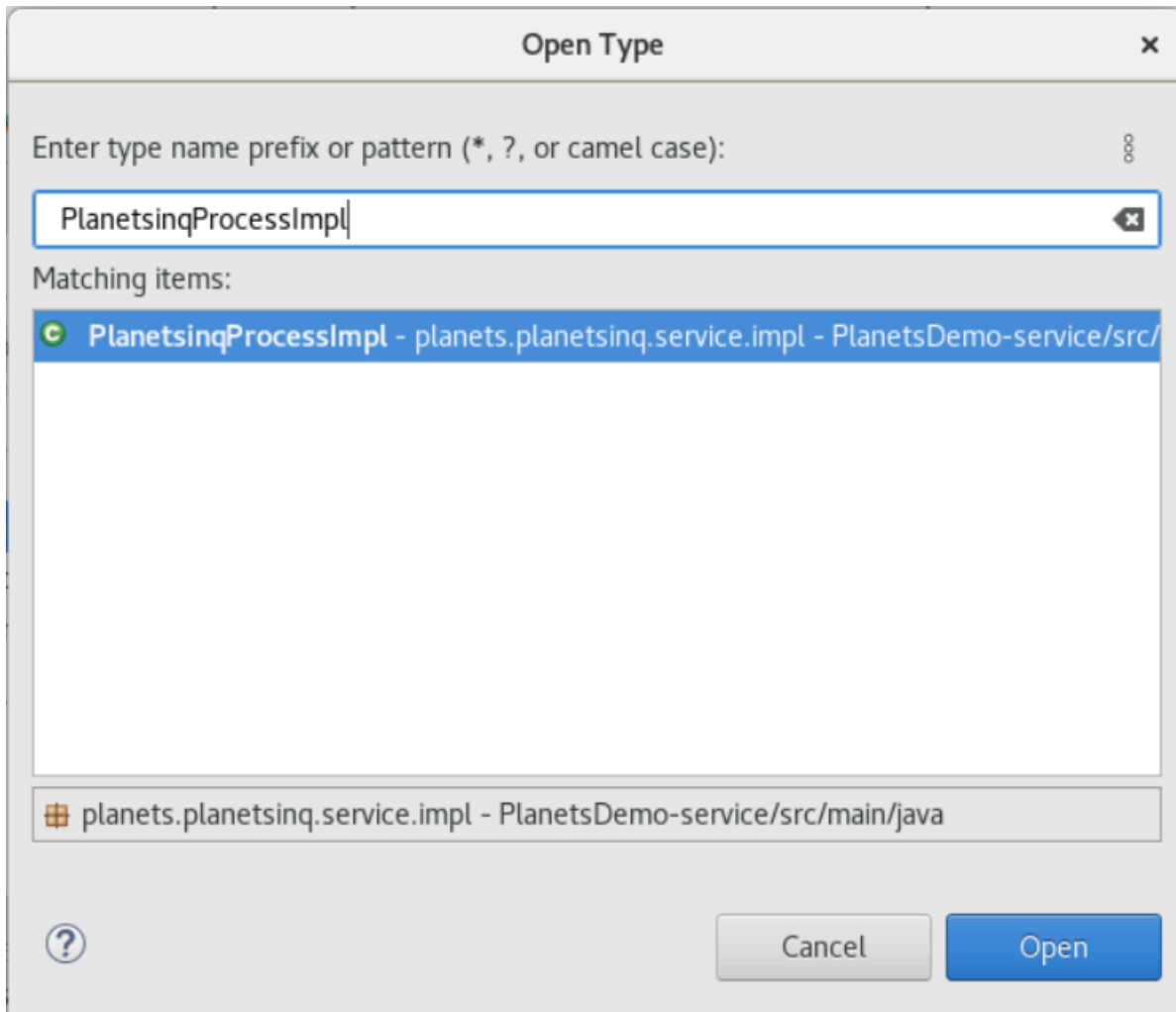
Planet name. . . . . :EARTH

Mass (x10^24kg). . . . . 0005.970
Diameter (km). . . . . 012756
Density (kg/m3). . . . . 5514
Length of day (h). . . . 0024.0
Dist. to sun (x10^6) . . 0149.6
Orbital period (days). 00365.2
Mean temperature (C) . . +015
Number of moons. . . . . 01
Has a ring system. . . . N
```

## Fase 10: Eseguire il debug dell'applicazione

In questo passaggio, esegui il test utilizzando le funzionalità di debug standard di Eclipse. Queste funzionalità sono disponibili quando si lavora su un'applicazione modernizzata.

1. Per aprire la classe di servizio principale, premi Ctrl + Shift + T. Quindi inserisci.  
`PlanetsinqProcessImpl`



2. Passa al `searchPlanet` metodo e inserisci lì un punto di interruzione.
3. Seleziona il nome del server e seleziona Riavvia in Debug.
4. Ripeti i passaggi precedenti. Cioè, accedi all'applicazione, inserisci il nome di un pianeta e premi Invio.

Eclipse interromperà l'applicazione nel `searchPlanet` metodo. Ora puoi esaminarlo.

## Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale in modo da non incorrere in costi aggiuntivi. Completa questa procedura:

- Se l'applicazione Planets è ancora in esecuzione, interrompila.
- Elimina il database in cui hai creato [Fase 1: crea un database](#). Per ulteriori informazioni, consulta la sezione relativa all'[eliminazione di un'istanza database](#).



# Ripiattaforma delle applicazioni con Micro Focus

Questa sezione descrive ogni fase del processo di ripiattaforma. Descrive tutte le attività e include informazioni sulla configurazione e il funzionamento del runtime di modernizzazione del AWS mainframe su Amazon EC2.

## Argomenti

- [Configurazione di Micro Focus Runtime \(su Amazon EC2\)](#)
- [Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#)
- [Tutorial: configurare Enterprise Analyzer su 2.0 AppStream](#)
- [Tutorial: configurare Micro Focus Enterprise Developer sulla AppStream versione 2.0](#)
- [Configurazione dell'automazione per le sessioni di streaming per sviluppatori Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#)
- [Visualizza i set di dati come tabelle e colonne in Enterprise Developer](#)
- [Tutorial: Usare i modelli con Micro Focus Enterprise](#)
- [Tutorial: configurazione della build Micro Focus per l'applicazione BankDemo di esempio](#)
- [Tutorial: configurazione di una pipeline CI/CD da utilizzare con Micro Focus Enterprise Developer](#)
- [Utilità Batch nella modernizzazione del AWS mainframe](#)

## Configurazione di Micro Focus Runtime (su Amazon EC2)

AWS Mainframe Modernization fornisce diverse Amazon Machine Images (AMI) che includono prodotti con licenza Micro Focus. Queste AMI consentono di effettuare rapidamente il provisioning di istanze Amazon Elastic Compute Cloud (Amazon EC2) per supportare ambienti Micro Focus controllati e gestiti da te. Questo argomento fornisce i passaggi necessari per accedere e avviare queste AMI. L'utilizzo di queste AMI è del tutto facoltativo e non è necessario per completare i tutorial di questa guida per l'utente.

## Argomenti

- [Prerequisiti](#)
- [Crea l'endpoint Amazon VPC per Amazon S3](#)
- [Richiedi l'aggiornamento della lista consentita per l'account](#)

- [Creazione del AWS Identity and Access Management ruolo](#)
- [Concedi a License Manager le autorizzazioni richieste](#)
- [Abbonati ad Amazon Machine Images](#)
- [Avvia un'istanza Micro Focus per la AWS modernizzazione del mainframe](#)
- [Subnet o VPC senza accesso a Internet](#)
- [Risoluzione dei problemi relativi alla licenza](#)

## Prerequisiti

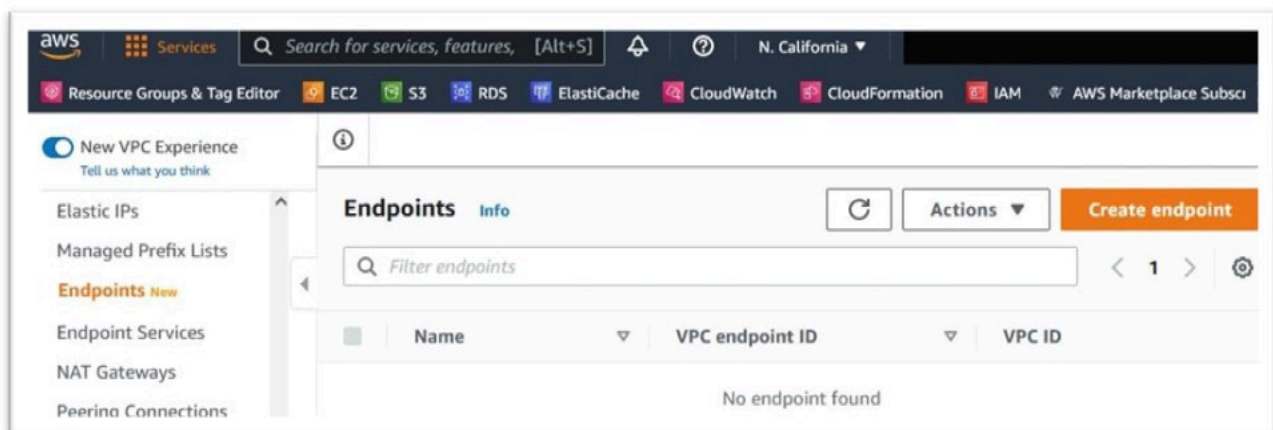
Assicurati di soddisfare i seguenti prerequisiti.

- Accesso dell'amministratore all'account in cui verranno create le istanze Amazon EC2.
- Identifica Regione AWS dove verranno create le istanze Amazon EC2 e verifica la disponibilità del servizio di modernizzazione del AWS mainframe. [Vedi Servizi per regione AWS](#) . Assicurati di scegliere una regione in cui il servizio è disponibile.
- Identifica l'Amazon Virtual Private Cloud (Amazon VPC) in cui verranno create le istanze Amazon EC2.

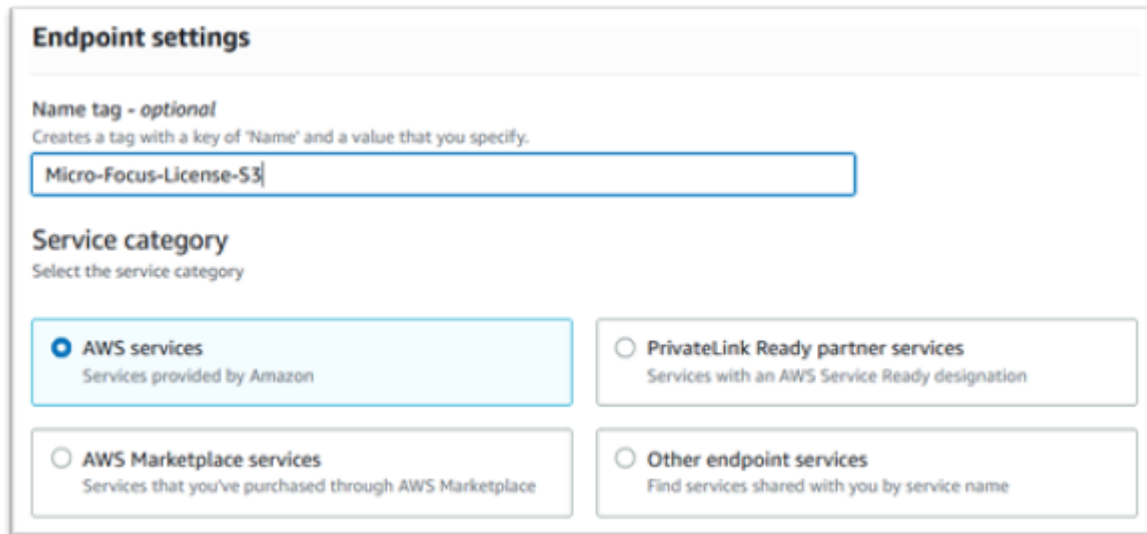
## Crea l'endpoint Amazon VPC per Amazon S3

In questa sezione, crei un endpoint Amazon VPC da utilizzare con Amazon S3.

1. Accedi ad Amazon VPC in. AWS Management Console
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Seleziona Crea endpoint.



- Inserisci un tag con il nome significativo, ad esempio: «Micro-Focus-License-S3».
- Scegli i servizi AWS come categoria di servizi.



**Endpoint settings**

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

Micro-Focus-License-S3

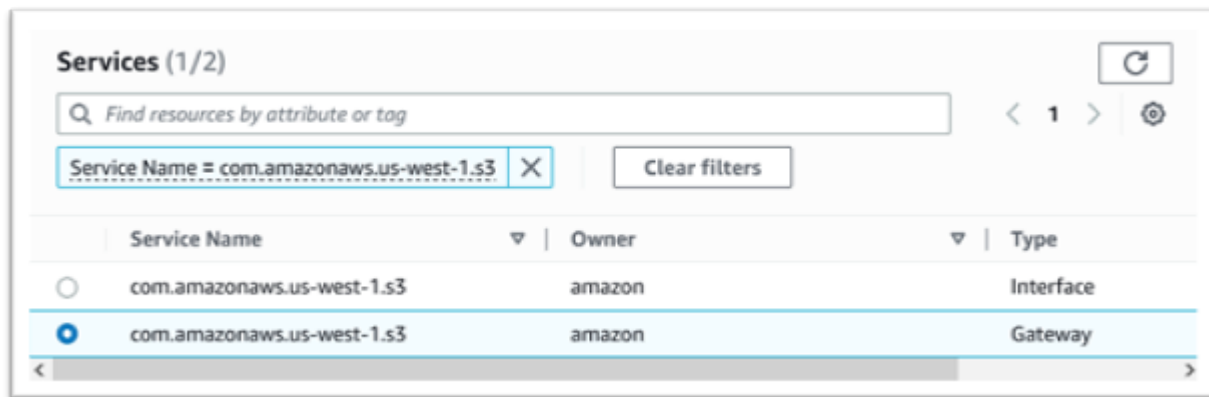
**Service category**  
Select the service category

- AWS services**  
Services provided by Amazon
- PrivateLink Ready partner services**  
Services with an AWS Service Ready designation
- AWS Marketplace services**  
Services that you've purchased through AWS Marketplace
- Other endpoint services**  
Find services shared with you by service name

- In Servizi cerca il servizio Amazon S3 Gateway: com.amazonaws. [regione] .s3.

Per us-west-1 questo sarebbe: com.amazonaws.us-west-1.s3

- Scegli il servizio Gateway.



**Services (1/2)**

Find resources by attribute or tag

Service Name = com.amazonaws.us-west-1.s3 X Clear filters

	Service Name	Owner	Type
<input type="radio"/>	com.amazonaws.us-west-1.s3	amazon	Interface
<input checked="" type="radio"/>	com.amazonaws.us-west-1.s3	amazon	Gateway

- Per VPC scegli il VPC che utilizzerai.

9. Scegli tutte le tabelle di routing per il VPC.

<input checked="" type="checkbox"/>	Name	Route Table ID
<input checked="" type="checkbox"/>	Modernization-rtb-public	rtb-██████████ (Modernization-rtb-public)
<input checked="" type="checkbox"/>	Modernization-rtb-private2-us-west-1c	rtb-██████████ (Modernization-rtb-private2-us-wes...)
<input checked="" type="checkbox"/>	Modernization-rtb-private1-us-west-1b	rtb-██████████ (Modernization-rtb-private1-us-west...)

10. In Policy scegli Accesso completo.

11. Scegliere Create Endpoint (Crea endpoint).

## Richiedi l'aggiornamento della lista consentita per l'account

Collabora con il tuo AWS rappresentante per inserire il tuo account nella lista consentita per le AMI di modernizzazione del AWS mainframe. Fornisci le seguenti informazioni:

- L' Account AWS ID.

- Il Regione AWS luogo in cui è stato creato l'endpoint Amazon VPC.
- L'ID endpoint Amazon VPC Amazon S3 creato in. [Crea l'endpoint Amazon VPC per Amazon S3](#)  
Questo è l'**vpce-xxxxxxxxxxxxxxxxx**id per com.amazonaws. [regione] .s3 Gateway Endpoint.
- Il numero di licenze richieste in tutte le istanze di Micro Focus Enterprise Suite AMI Amazon EC2.

È richiesta una licenza per core della CPU (per 2 vCPU per la maggior parte delle istanze Amazon EC2).

[Per ulteriori informazioni, consulta Ottimizzare le opzioni della CPU.](#)

Il numero richiesto potrà essere modificato in futuro tramite AWS.

#### Note

Il AWS rappresentante deve aprire il ticket di supporto per la richiesta Allowlist. Non può essere richiesto direttamente e il completamento della richiesta potrebbe richiedere diversi giorni.

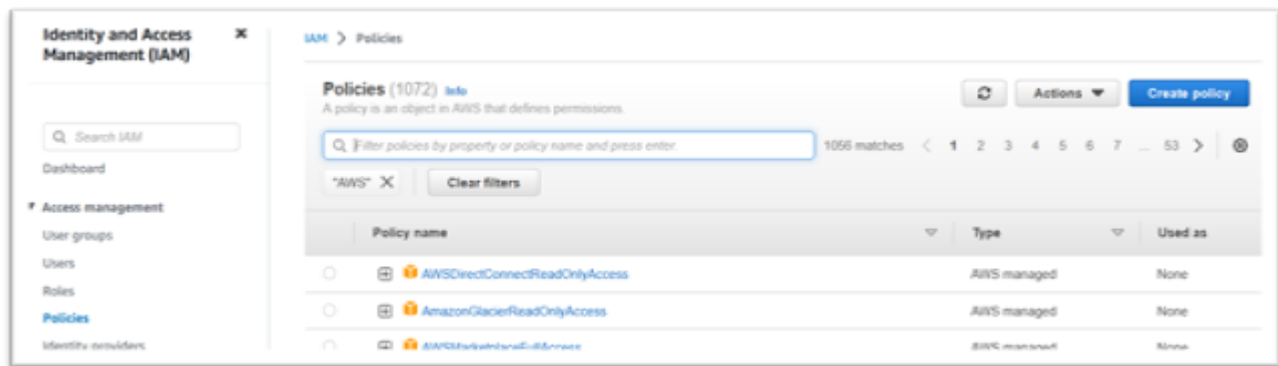
## Creazione del AWS Identity and Access Management ruolo

Crea una AWS Identity and Access Management policy e un ruolo da utilizzare per le istanze Amazon EC2 di modernizzazione del AWS mainframe. La creazione del ruolo tramite la console IAM creerà un profilo di istanza associato con lo stesso nome. L'assegnazione di questo profilo di istanza alle istanze Amazon EC2 consente l'assegnazione delle licenze Micro Focus. Per ulteriori informazioni sui profili di istanza, consulta [Usare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.](#)

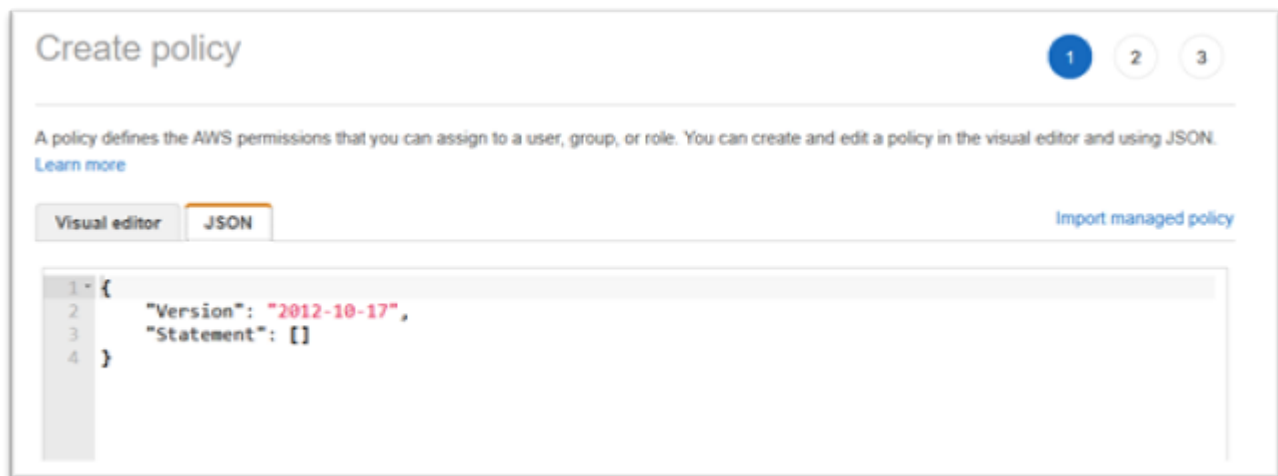
### Creare una policy IAM

Una policy IAM viene prima creata e poi associata al ruolo.

1. Passa AWS Identity and Access Management a AWS Management Console.
2. Scegli Politiche e poi Crea politica.



3. Seleziona la scheda JSON.



4. Sostituisci us-west-1 il codice JSON seguente con quello Regione AWS in cui è stato definito l'endpoint Amazon S3, quindi copia e incolla il codice JSON nell'editor delle policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "S3WriteObject",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-supernova-marketplace-us-west-1-prod/*"
      ]
    },
    {
      "Sid": "OtherRequiredActions",
```

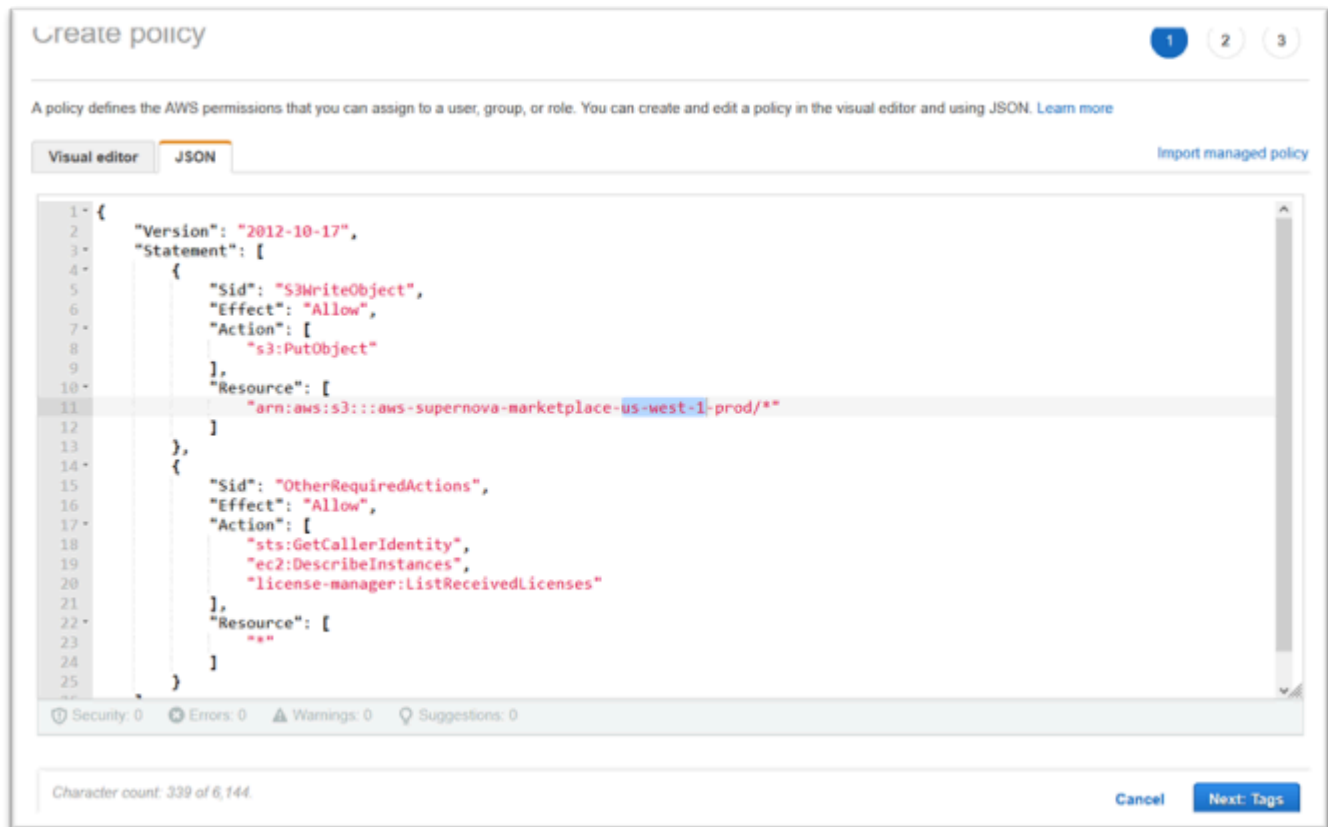
```

    "Effect": "Allow",
    "Action": [
      "sts:GetCallerIdentity",
      "ec2:DescribeInstances",
      "license-manager:ListReceivedLicenses"
    ],
    "Resource": [
      "*"
    ]
  }
]
}

```

### Note

Le azioni sotto il Sid `OtherRequiredActions` non supportano le autorizzazioni a livello di risorsa e devono essere specificate nell'elemento `resource`. \*



## 5. Scegliere Successivo: Tag.

**Create policy** 1 2 3

**Add tags - optional**  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

**Add tag**

You can add up to 50 more tags.

[Cancel](#) [Previous](#) [Next: Review](#)

6. Facoltativamente, inserisci qualsiasi tag, quindi scegli Avanti: Revisione.
7. Inserisci un nome per la politica, ad esempio «Micro-Focus-Licensing-Policy». Facoltativamente, inserisci una descrizione, ad esempio «Un ruolo che include questa policy deve essere associato a ciascuna istanza Amazon EC2 per la modernizzazione del AWS mainframe».

**Create policy** 1 2 3

**Review policy**

**Name\***   
Use alphanumeric and '+=, @, \_' characters. Maximum 128 characters.

**Description**   
Maximum 1000 characters. Use alphanumeric and '+=, @, \_' characters.

**Summary**

Service	Access level	Resource	Request condition
Allow (4 of 369 services) <a href="#">Show remaining 365</a>			
EC2	Limited: List	All resources	None
License Manager	Limited: List	All resources	None
S3	Limited: Write	BucketName   string like   aws-supernova-marketplace-us-west-1-prod, ObjectPath   string like   All	None
STS	Limited: Read	All resources	None

**Tags**

Key	Value
No tags associated with the resource.	

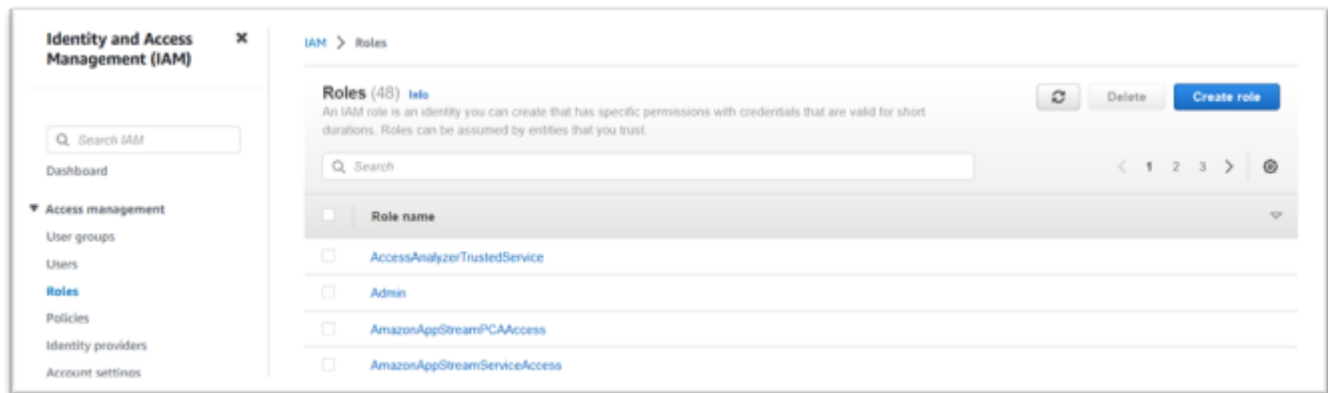
\* Required [Cancel](#) [Previous](#) [Create policy](#)

8. Scegliere Create Policy (Crea policy).

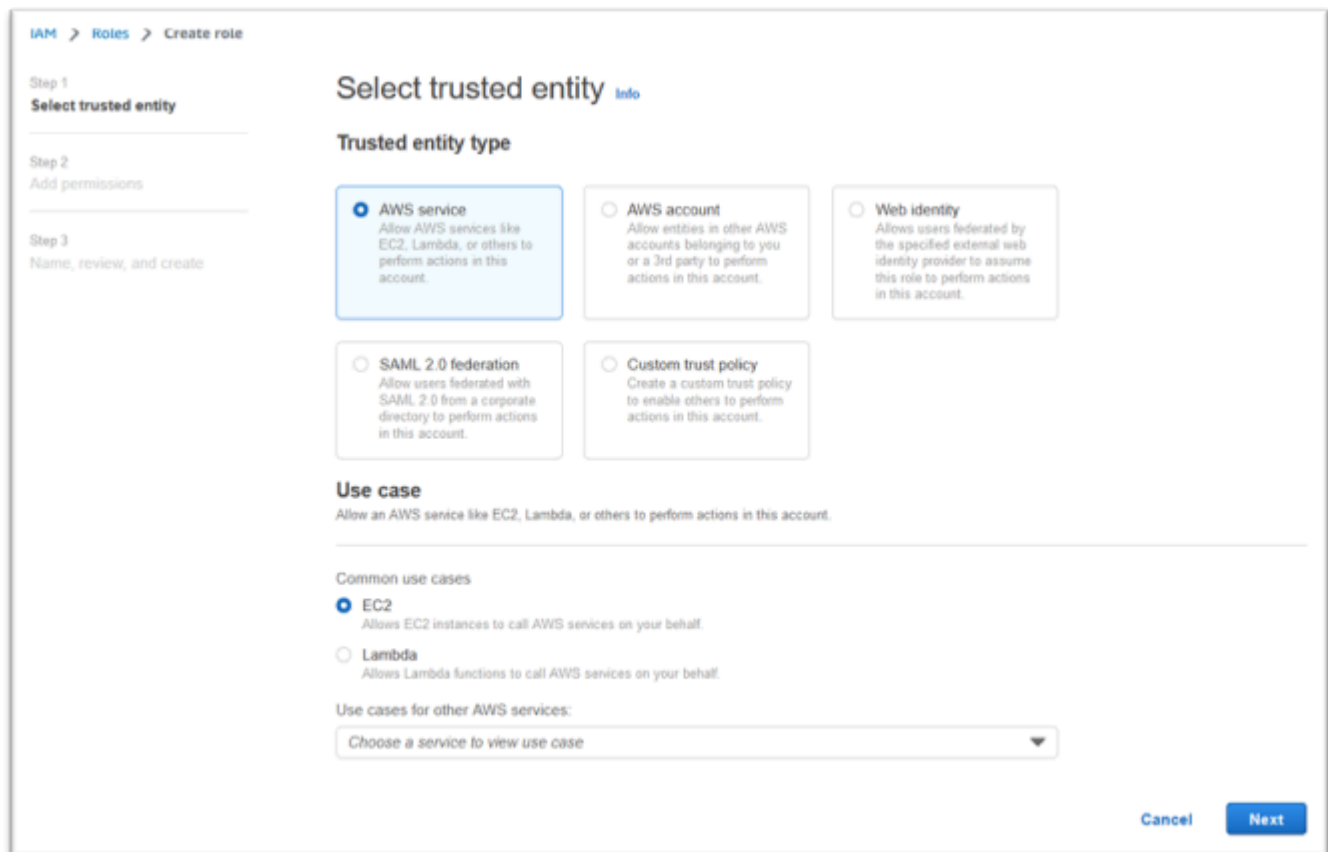


## Crea il ruolo IAM

1. Passa a IAM in AWS Management Console.
2. Scegli Ruoli e poi Crea ruolo.

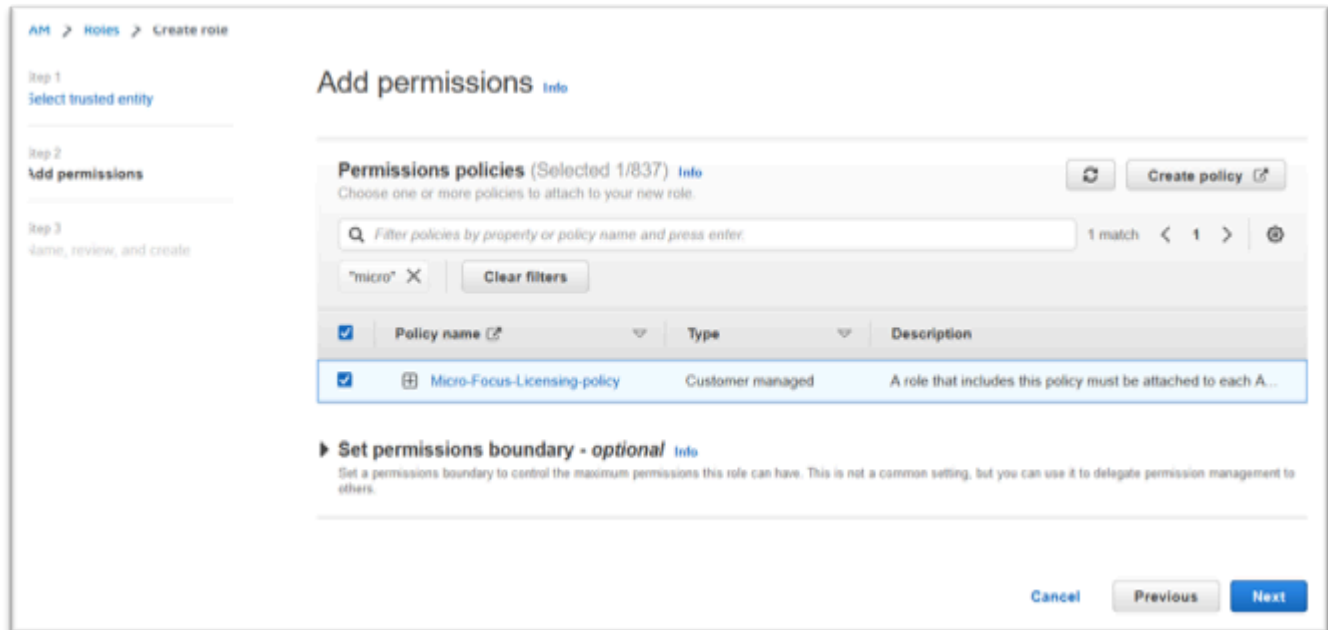


3. Lascia il tipo di entità affidabile come AWS servizio e scegli il caso d'uso comune di EC2.



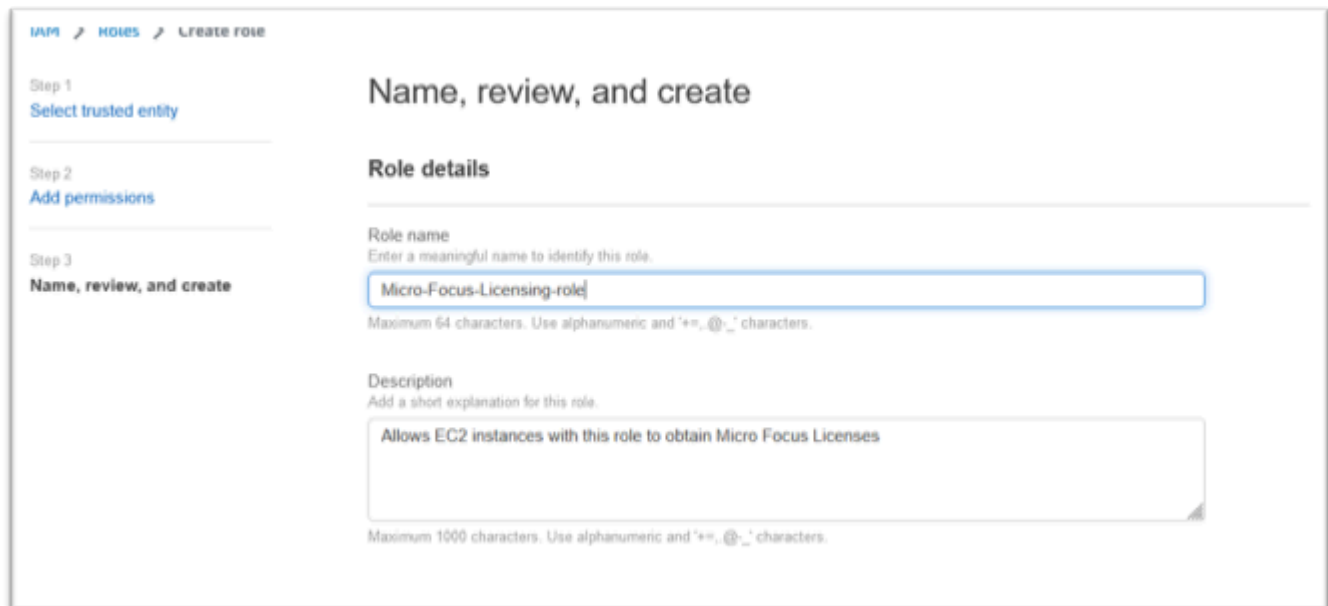
4. Seleziona Avanti.
5. Inserisci «Micro» nel filtro e premi invio per applicare il filtro.
6. Scegli la politica appena creata, ad esempio la «Micro-Focus-Licensing-Policy».

## 7. Seleziona Avanti.



8. Immettete il nome del ruolo, ad esempio «Micro-Focus-Licensing-Role».

9. Sostituisci la descrizione con una tua, ad esempio «Consente alle istanze Amazon EC2 con questo ruolo di ottenere licenze Micro Focus».



10. Nella fase 1: Seleziona entità attendibili, esamina il codice JSON e conferma che abbia i seguenti valori:

```
{
```

```

"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "sts:AssumeRole"
    ],
    "Principal": {
      "Service": [
        "ec2.amazonaws.com"
      ]
    }
  }
]
}

```

### Note

L'ordine di Effetto, Azione e Principal non è significativo.

11. Verifica che la Fase 2: Aggiungi autorizzazioni mostri la tua politica di gestione delle licenze.

Step 2: Add permissions Edit

Permissions policy summary

Policy name <a href="#">↗</a>	Type	Attached as
Micro-Focus-Licensing-policy	Customer managed	Permissions policy

Tags

**Add tags - optional** [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Previous Create role

12. Scegli Crea ruolo.

Una volta completata la richiesta della lista consentita, continua con i seguenti passaggi.

## Concedi a License Manager le autorizzazioni richieste

1. Vai AWS License Manager a. AWS Management Console

Management & Governance

# AWS License Manager

Manage, discover, and report software license usage

AWS License Manager offers multiple ways to track license usage across your environments. Get started with user-based licenses, granted licenses, self managed licenses, or seller issued licenses.

### How it works

1. Define rules for your licensed software
2. Attach licensing rules during launch and proactively control usage
3. Search inventory and track licenses brought in past launch
4. Use alerts to control and centrally manage licenses across all AWS accounts and on-premises

### Get started

Set rules and manage third-party licenses proactively

[Start using AWS License Manager](#)

### Pricing

There is no additional charge for AWS License Manager.

For information about relevant AWS services, see the following pricing sections:

- [Amazon pricing](#)
- [Amazon EC2 pricing](#)
- [Amazon EBS pricing](#)
- [Amazon Systems Manager pricing](#)
- [Amazon SNS pricing](#)

2. Scegli Inizia a usare AWS License Manager.
3. Se vedi il seguente pop-up, visualizza i dettagli, quindi seleziona la casella di controllo e premi Concedi autorizzazioni.

**IAM permissions (one-time setup)**

AWS License Manager requires permissions to manage licenses used by resources.

I grant AWS License Manager the required permissions

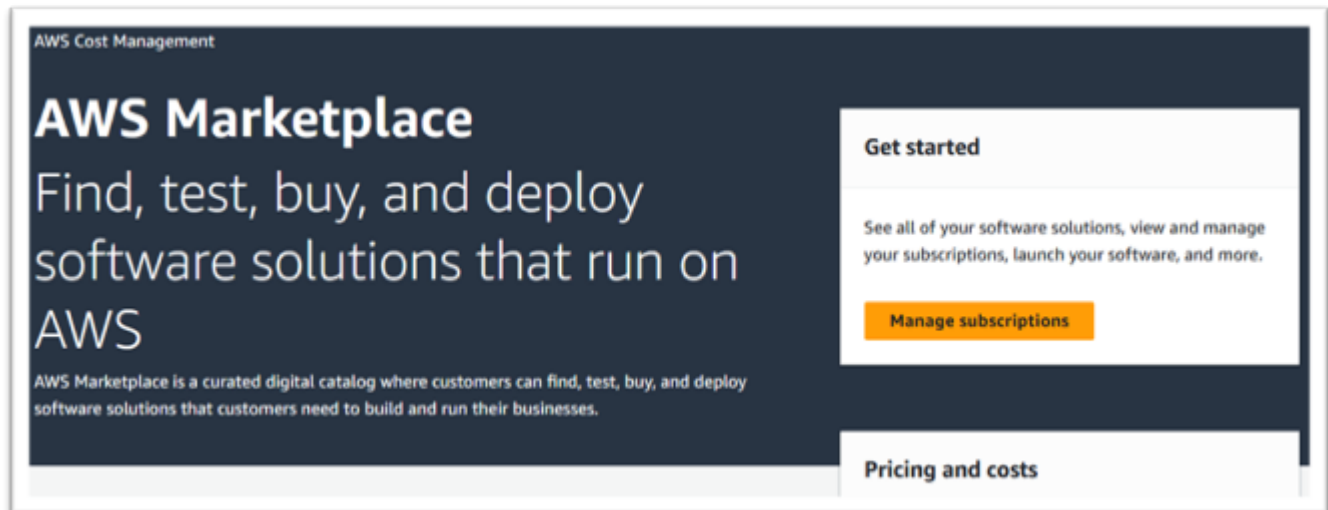
[View details](#)

[Cancel](#) [Grant permissions](#)

## Abbonati ad Amazon Machine Images

Dopo esserti abbonato a un Marketplace AWS prodotto, puoi avviare un'istanza dall'AMI del prodotto.

1. Vai a Marketplace AWS Abbonamenti in. AWS Management Console
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).



3. Copia e incolla uno dei seguenti link nella barra degli indirizzi del browser.

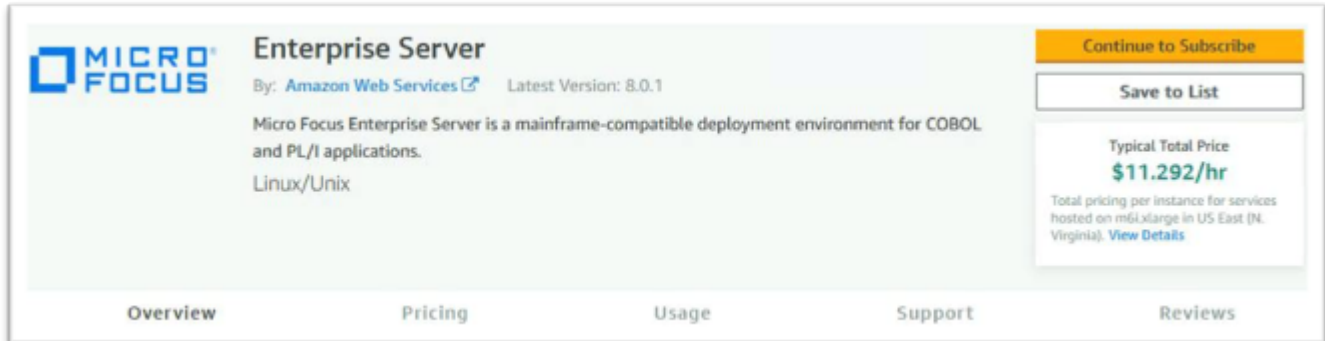
### Note

Scegli un link solo per uno dei prodotti che sei stato autorizzato a utilizzare.

- Server aziendale: <https://aws.amazon.com/marketplace/pp/prodview-g5emev63l7blc>
- Enterprise Server per Windows: <https://aws.amazon.com/marketplace/pp/prodview-lwybsiyikbhc2>
- Sviluppatore aziendale: <https://aws.amazon.com/marketplace/pp/prodview-77qmpr42yzxwk>
- Sviluppatore aziendale con Visual Studio 2022: <https://aws.amazon.com/marketplace/pp/prodview-m4l3lqiszo6cm>
- Enterprise Analyzer: <https://aws.amazon.com/marketplace/pp/prodview-tttheylcmcihm>
- Enterprise Build Tools per Windows: <https://aws.amazon.com/marketplace/pp/prodview-2rw35bbt6uozl>
- Enterprise Stored Procedure: <https://aws.amazon.com/marketplace/pp/prodview-zoeyqnsdsj6ha>

- Stored Procedure aziendali con SQL Server 2019: <https://aws.amazon.com/marketplace/pp/prodview-ynfklquwubnz4>

4. Scegliere Continue to Subscribe (Continua la sottoscrizione).



**MICRO FOCUS** Enterprise Server  
By: [Amazon Web Services](#) Latest Version: 8.0.1

Micro Focus Enterprise Server is a mainframe-compatible deployment environment for COBOL and PL/I applications.  
Linux/Unix

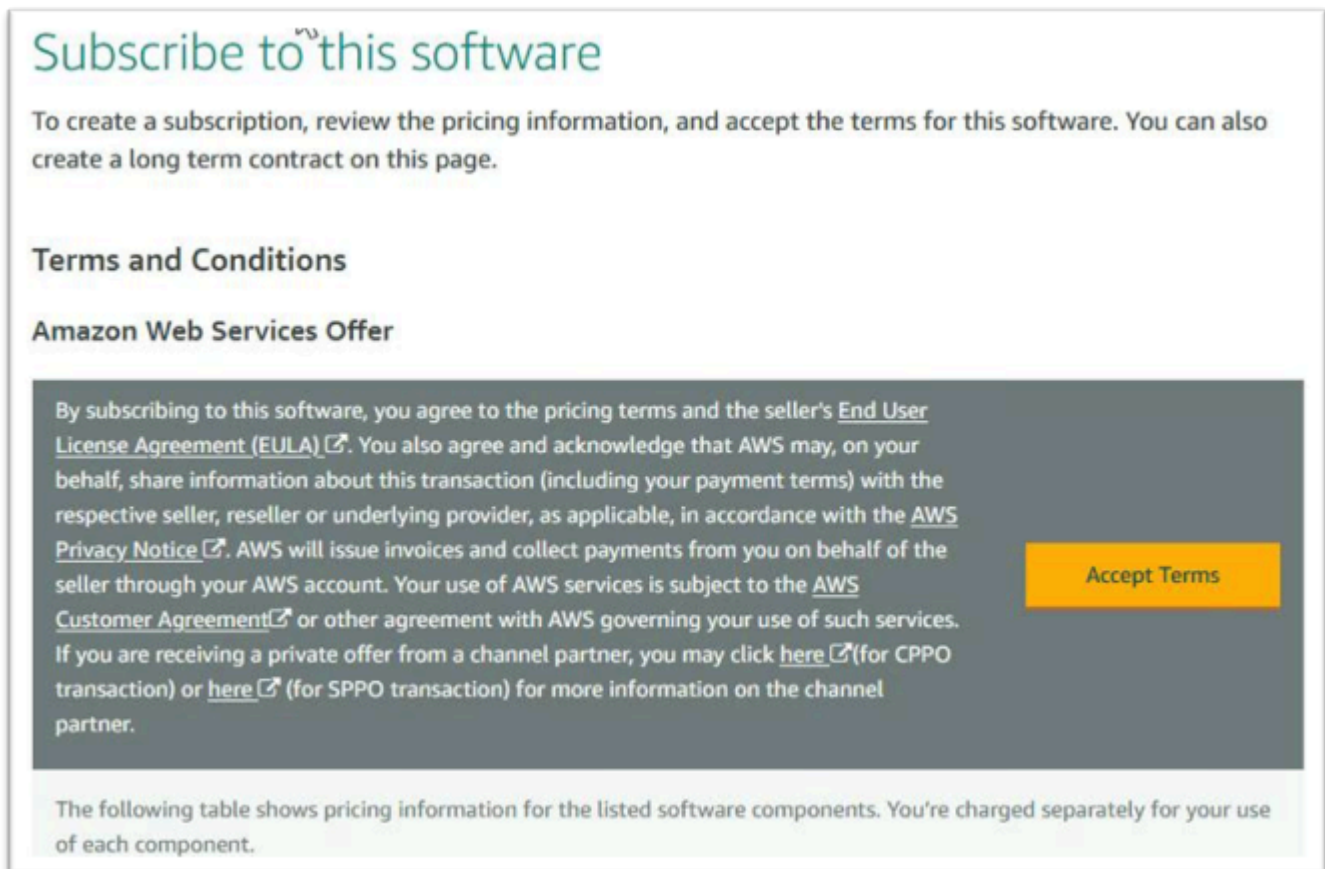
[Continue to Subscribe](#)

[Save to List](#)

Typical Total Price  
**\$11.292/hr**  
Total pricing per instance for services hosted on m6i.xlarge in US East (N. Virginia). [View Details](#)

Overview Pricing Usage Support Reviews

5. Se i Termini e condizioni sono accettabili, scegli Accetta termini.



## Subscribe to this software

To create a subscription, review the pricing information, and accept the terms for this software. You can also create a long term contract on this page.

### Terms and Conditions

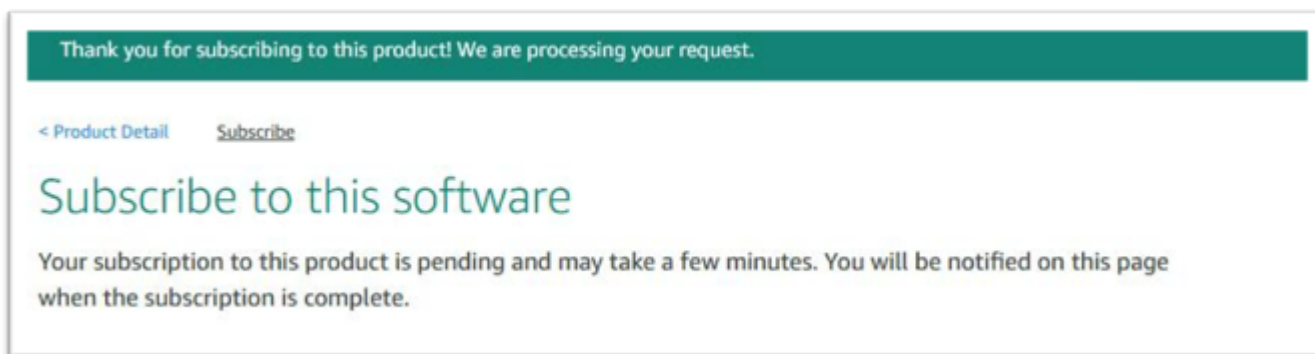
### Amazon Web Services Offer

By subscribing to this software, you agree to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You also agree and acknowledge that AWS may, on your behalf, share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services. If you are receiving a private offer from a channel partner, you may click [here](#) (for CPPO transaction) or [here](#) (for SPPO transaction) for more information on the channel partner.

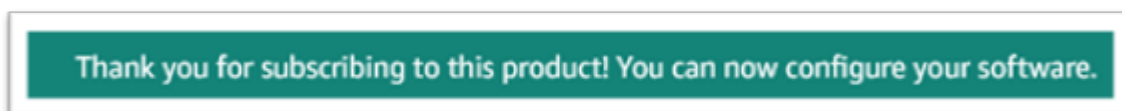
[Accept Terms](#)

The following table shows pricing information for the listed software components. You're charged separately for your use of each component.

6. L'elaborazione dell'abbonamento potrebbe richiedere alcuni minuti.



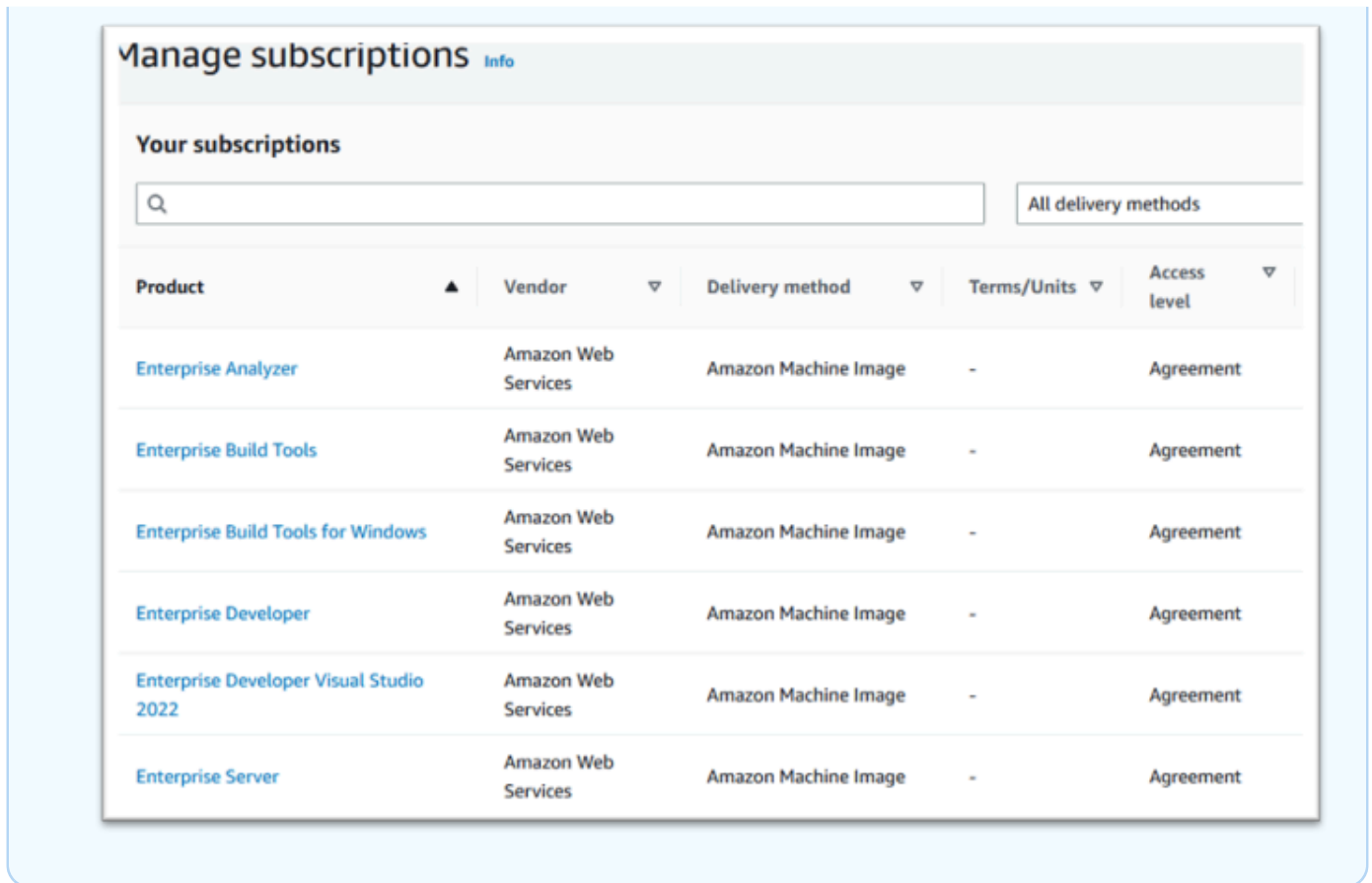
7. Dopo la visualizzazione del messaggio di ringraziamento, copia e incolla il link successivo dal passaggio 3 per continuare ad aggiungere abbonamenti.



8. Interrompi quando Manage subscriptions mostra tutte le AMI sottoscritte.

#### Note

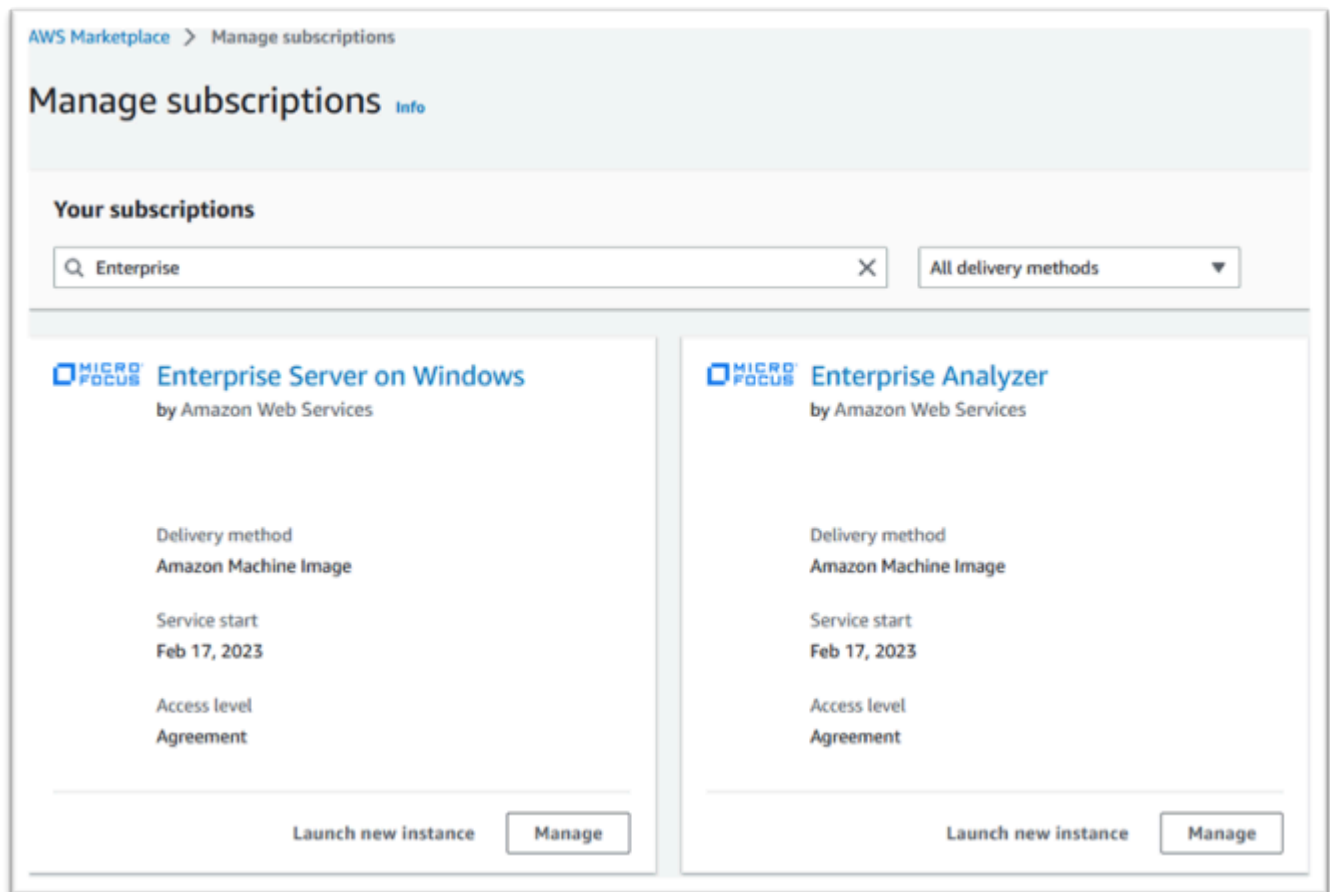
Le preferenze del pannello (icona a forma di ingranaggio) sono impostate per mostrare la visualizzazione come tabella.



## Avvia un'istanza Micro Focus per la AWS modernizzazione del mainframe

1. Vai a Marketplace AWS Abbonamenti in. AWS Management Console
2. Individua l'AMI da lanciare e scegli Launch New Instance.





3. Nella finestra di dialogo di avvio di una nuova istanza, assicurati che la regione consentita sia selezionata.
4. Premi Continua per avviare tramite EC2.

#### Note

L'esempio seguente mostra il lancio di un'AMI Enterprise Developer, ma il processo è lo stesso per tutte le AMI di modernizzazione del AWS mainframe.

The screenshot shows the 'Launch new instance' configuration page in the AWS Marketplace. The breadcrumb navigation at the top reads: 'AWS Marketplace > Manage subscriptions > Enterprise Developer > Launch new instance'. The main heading is 'Launch new instance'. Below this is a section titled 'Configure this software' with the instruction: 'Choose a fulfillment option below to select how you wish to deploy the software, then enter the information required to configure the deployment.' The configuration options are: 'Delivery method' set to '64-bit (x86) Amazon Machine Image', 'Software version' set to 'v8.0.1 (Oct 26, 2022)', and 'Region' set to 'us-west-1'. Below these is the 'AMI ID: ami-0f199167bc5fce009'. At the bottom right, there are two buttons: 'Cancel' and 'Continue to launch through EC2'.

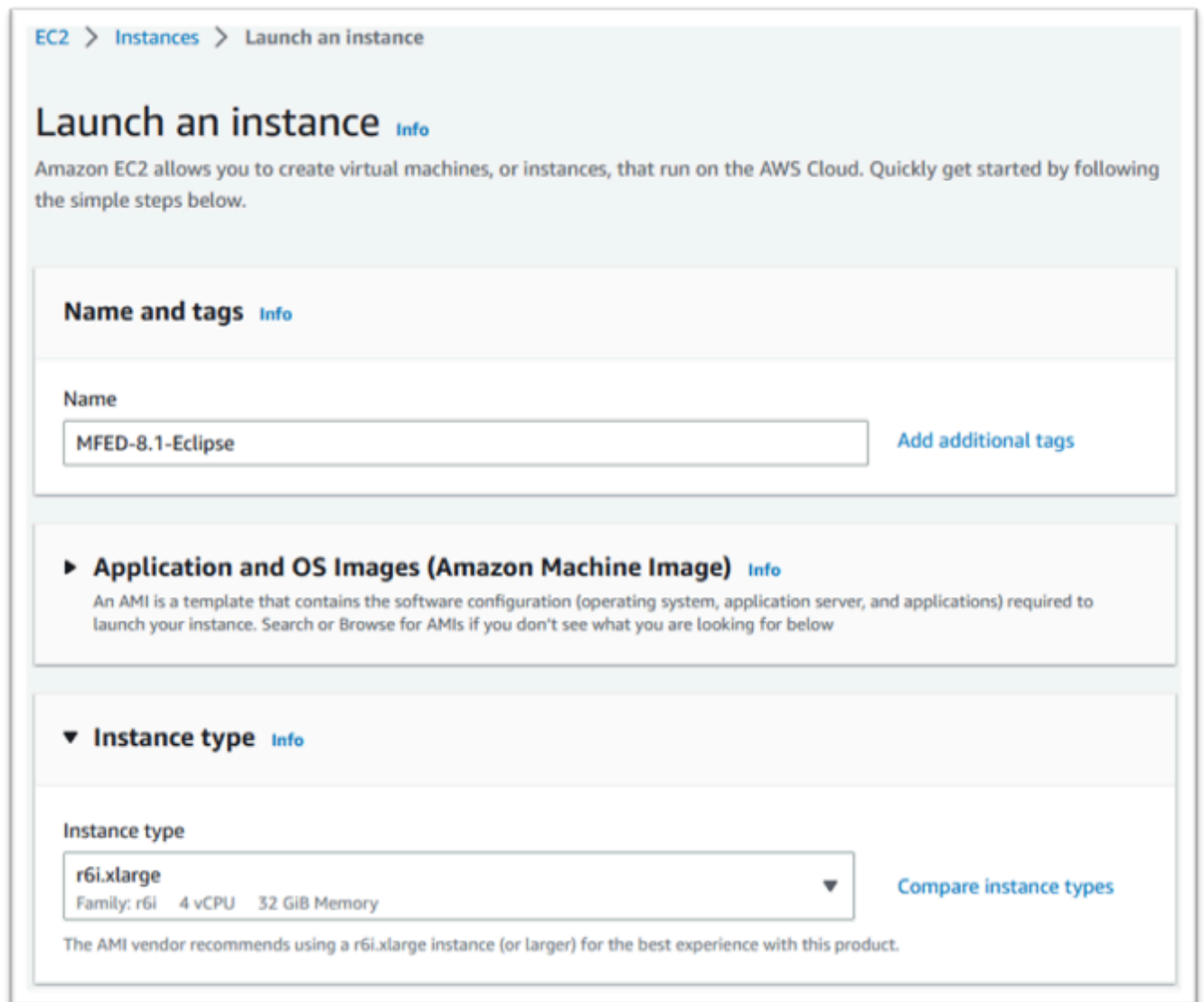
5. Inserisci un nome per il server.
6. Scegliere un tipo di istanza.

Il tipo di istanza selezionato deve essere determinato dai requisiti di prestazioni e costi del progetto. I seguenti sono i punti di partenza suggeriti:

- Per Enterprise Analyzer, un r6i.xlarge
- Per Enterprise Developer, un r6i.large
- Per un'istanza autonoma di Enterprise Server, un r6i.xlarge
- Per Micro Focus Performance Availability Cluster (PAC) con scale-out, un r6i.large

**Note**

La sezione Immagini dell'applicazione e del sistema operativo è stata compressa rispetto alla schermata.



EC2 > Instances > Launch an instance

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags [Info](#)

Name

 [Add additional tags](#)


7. Scegliete o create (e salvate) una coppia di chiavi (non mostrata).

Per ulteriori informazioni sulle coppie di chiavi per le istanze Linux, consulta [Coppie di chiavi Amazon EC2 e istanze Linux](#).

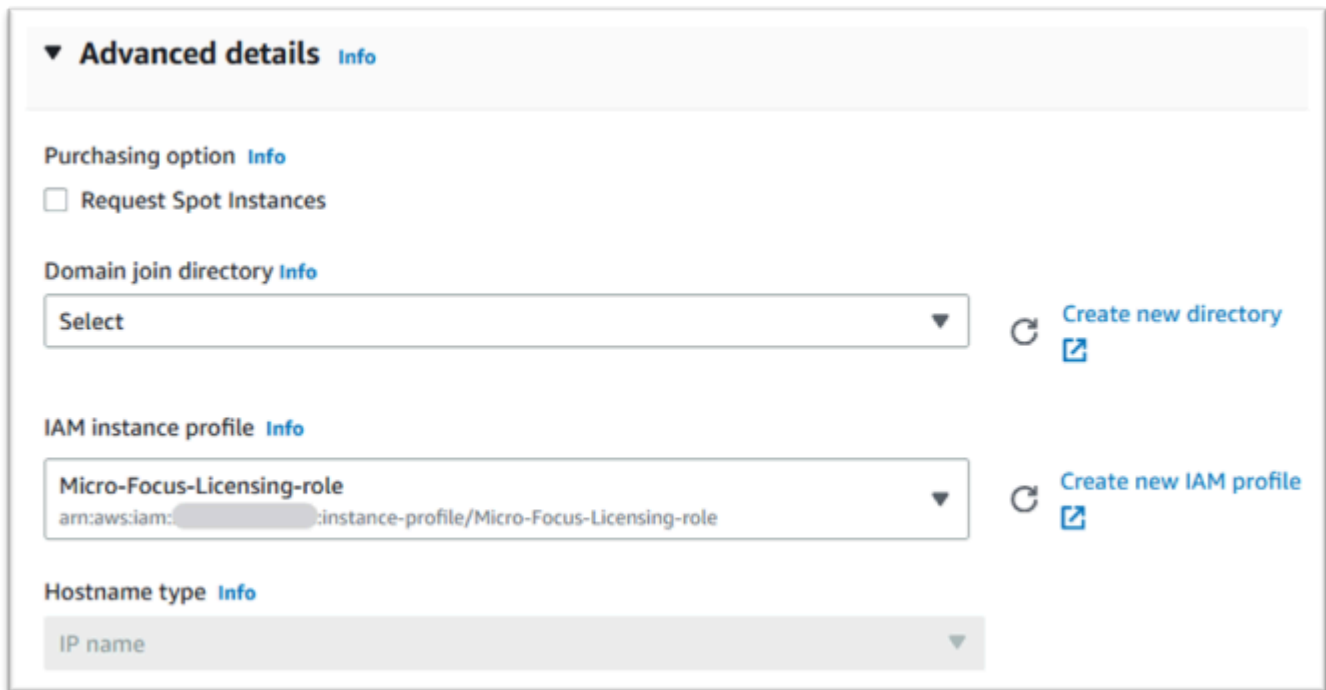
Per ulteriori informazioni sulle coppie di chiavi per le istanze Windows, consulta [Coppie di chiavi Amazon EC2 e istanze Windows](#).

8. Modifica le impostazioni di rete e scegli il VPC nell'elenco consentito e la sottorete appropriata.
9. Scegli o crea un gruppo di sicurezza. Se si tratta di un'istanza EC2 di Enterprise Server, è normale consentire al traffico TCP verso le porte 86 e 10086 per amministrare la configurazione Micro Focus.

10. Configura facoltativamente lo storage per l'istanza Amazon EC2.
11. Importante: espandi i dettagli di Advanced e nel profilo dell'istanza IAM scegli il ruolo di licenza creato in precedenza, ad esempio «Micro-Focus-Licensing-role».

 Note

Se questo passaggio viene saltato, dopo aver creato l'istanza puoi modificare il ruolo IAM dall'opzione Sicurezza del menu Azione per l'istanza EC2.



**Advanced details** [Info](#)

**Purchasing option** [Info](#)

Request Spot Instances

**Domain join directory** [Info](#)

Select [Create new directory](#)

**IAM instance profile** [Info](#)

Micro-Focus-Licensing-role  
arn:aws:iam:[:instance-profile/Micro-Focus-Licensing-role] [Create new IAM profile](#)

**Hostname type** [Info](#)

IP name

12. Esamina il riepilogo e invia Launch Instance.

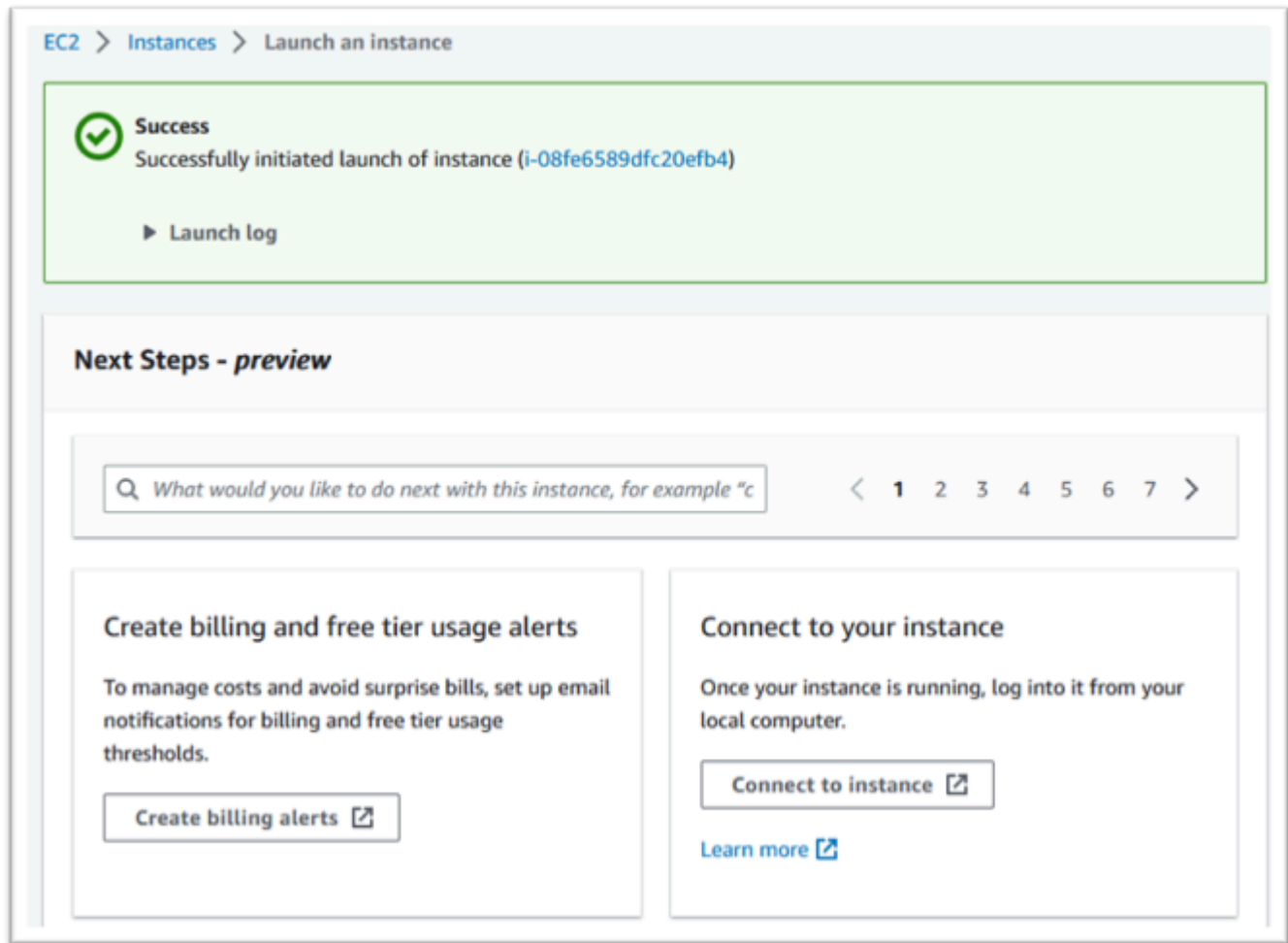
The screenshot shows the 'Summary' section of the AWS console. It includes a dropdown for the number of instances set to 1, a section for Software Image (AMI) with a link to read more and the ID ami-0f199167bc5fce009, a section for Virtual server type (instance type) set to r6i.xlarge, a section for Firewall (security group) set to default, and a section for Storage (volumes) set to 1 volume(s) - 100 GiB. A blue information box at the bottom states: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.' At the bottom right is a yellow 'Launch instance' button, and at the bottom left is a 'Cancel' button.

13. L'avvio dell'istanza avrà esito negativo se viene scelto un tipo di server virtuale non valido.

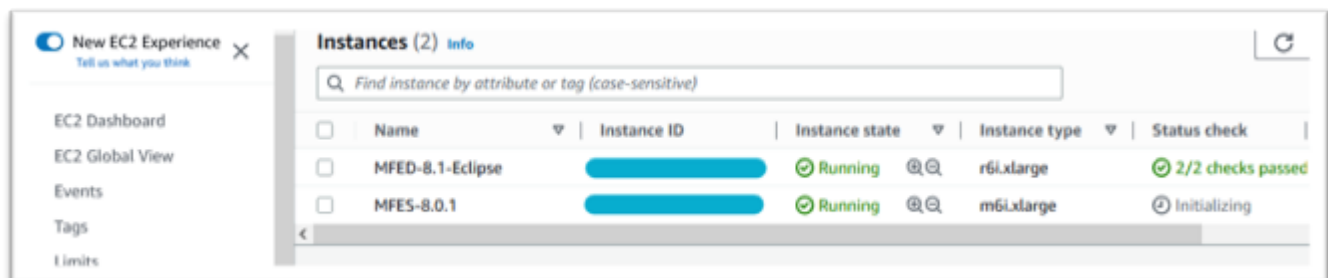
In tal caso, scegli Modifica configurazione dell'istanza e modifica il tipo di istanza.

The screenshot shows the 'Launching instance' progress bar. It displays the text 'Please wait while we launch your instance. Do not close your browser while this is loading.' Below this is a progress bar for 'Subscribing to Marketplace AMI' which is at 73%. A 'Details' link is visible below the progress bar.

14. Una volta visualizzato il messaggio «Success», scegli Connetti all'istanza per ottenere i dettagli della connessione.



15. In alternativa, accedi a EC2 nel AWS Management Console.  
 16. Scegli Istanze per visualizzare lo stato della nuova istanza.



## Subnet o VPC senza accesso a Internet

Apporta queste modifiche aggiuntive se la sottorete o il VPC non dispone di accesso a Internet in uscita.

Il gestore delle licenze richiede l'accesso ai seguenti servizi AWS:

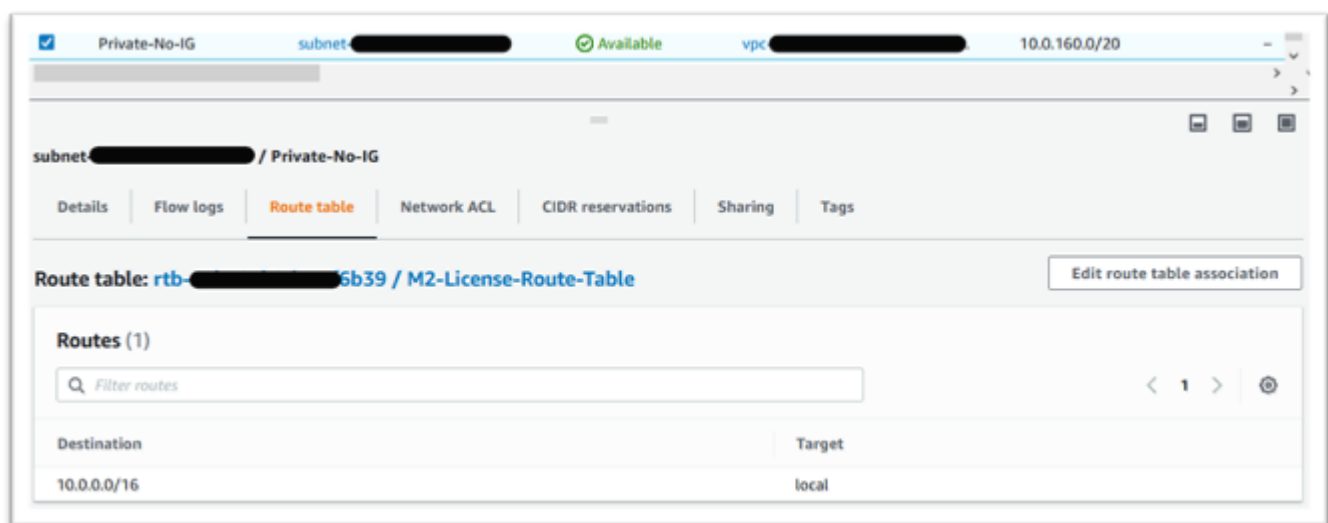
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ec2`
- `com.amazonaws.region.license-manager`
- `com.amazonaws.region.sts`

I passaggi precedenti hanno definito `com.amazonaws.servizio region.s3` come endpoint gateway. Questo endpoint necessita di una voce nella tabella di routing per tutte le sottoreti senza accesso a Internet.

I tre servizi aggiuntivi verranno definiti come endpoint di interfaccia.

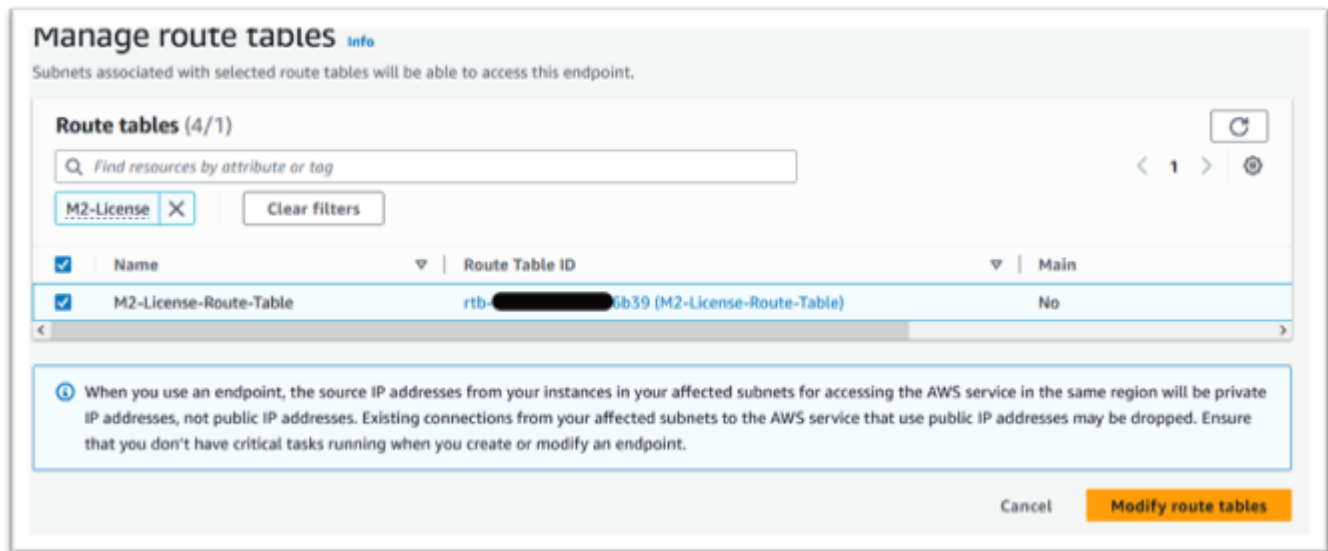
### Aggiungi la voce della tabella Route per l'endpoint Amazon S3

1. Vai a VPC in AWS Management Console e scegli Subnet.
2. Scegli la sottorete in cui verranno create le istanze Amazon EC2 e scegli la scheda Route Table.
3. Nota alcune cifre finali dell'id della tabella Route. Ad esempio, il 6b39 nell'immagine qui sotto.



4. Scegli Endpoints dal pannello di navigazione.

5. Scegli l'endpoint creato in precedenza e poi Gestisci le tabelle dei percorsi, dalla scheda Tabelle degli itinerari per l'endpoint o dal menu a discesa Azioni.
6. Scegli la tabella dei percorsi utilizzando le cifre identificate in precedenza e premi Modifica le tabelle degli itinerari.



## Definire il gruppo di sicurezza richiesto

I servizi Amazon EC2 e License Manager comunicano tramite HTTPS tramite la porta 443. AWS STS Questa comunicazione è bidirezionale e richiede regole in entrata e in uscita per consentire all'istanza di comunicare con i servizi.

1. Accedi ad Amazon VPC in AWS Management Console
2. Individua i gruppi di sicurezza nella barra di navigazione e scegli Crea gruppo di sicurezza.
3. Inserisci il nome e la descrizione del gruppo di sicurezza, ad esempio «HTTPS in entrata-uscita».
4. Premi la X nell'area di selezione del VPC per rimuovere il VPC predefinito e scegli il VPC che contiene l'endpoint S3.
5. Aggiungi una regola in entrata che consenta il traffico TCP sulla porta 443 da qualsiasi luogo.

### Note

Le regole in entrata (e in uscita) possono essere ulteriormente limitate limitando la fonte. Per ulteriori informazioni, consulta [Controlla il traffico verso AWS le tue risorse utilizzando i gruppi di sicurezza](#) nella Amazon VPC User Guide.



The screenshot displays the AWS Security Groups console interface. It is divided into two main sections: 'Basic details' and 'Inbound rules'.

**Basic details:**

- Security group name:** Inbound-Outbound HTTPS. A note below states: 'Name cannot be edited after creation.'
- Description:** Allow HTTPS traffic on port 443.
- VPC:** A dropdown menu showing a selected VPC.

**Inbound rules:**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	443	Anywh... 0.0.0.0/0	HTTPS traffic

Buttons for 'Add rule' and 'Delete' are visible.

6. Premi Crea gruppo di sicurezza.

## Crea gli endpoint del servizio

Ripeti questo processo tre volte, una volta per ogni servizio.

1. Accedi ad Amazon VPC in AWS Management Console e scegli Endpoints.
2. Premi Crea endpoint.
3. Inserisci un nome, ad esempio «Micro-Focus-License-EC2», «Micro-Focus-License-STS» o «Micro-Focus-License-Manager».
4. Scegli la categoria di servizi AWS.

**Endpoint settings**

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

Micro-Focus-License-EC2

**Service category**  
Select the service category

- AWS services**  
Services provided by Amazon
- PrivateLink Ready partner services**  
Services with an AWS Service Ready designation
- AWS Marketplace services**  
Services that you've purchased through AWS Marketplace
- Other endpoint services**  
Find services shared with you by service name

5. In Servizi cerca il servizio di interfaccia corrispondente, che è uno dei seguenti:

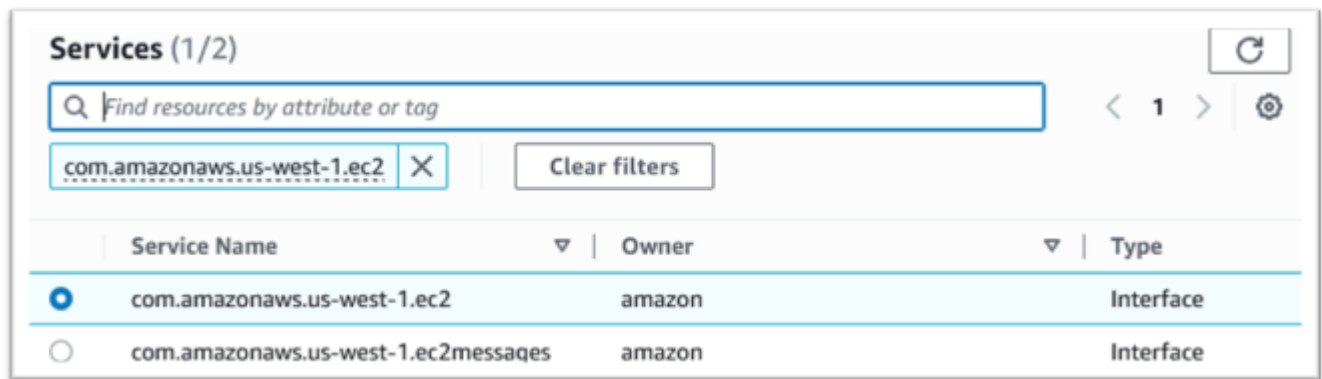
- «com.amazonaws. *regione* .ec2»
- «com.amazonaws. *regione* .sts»
- «com.amazonaws. *regione* .license-manager»

Per esempio:

- «com.amazonaws.us-west-1.ec2»
- «com.amazonaws.us-west-1.sts»
- «com.amazonaws.us-west-1.license-manager»

6. Scegli il servizio di interfaccia corrispondente.

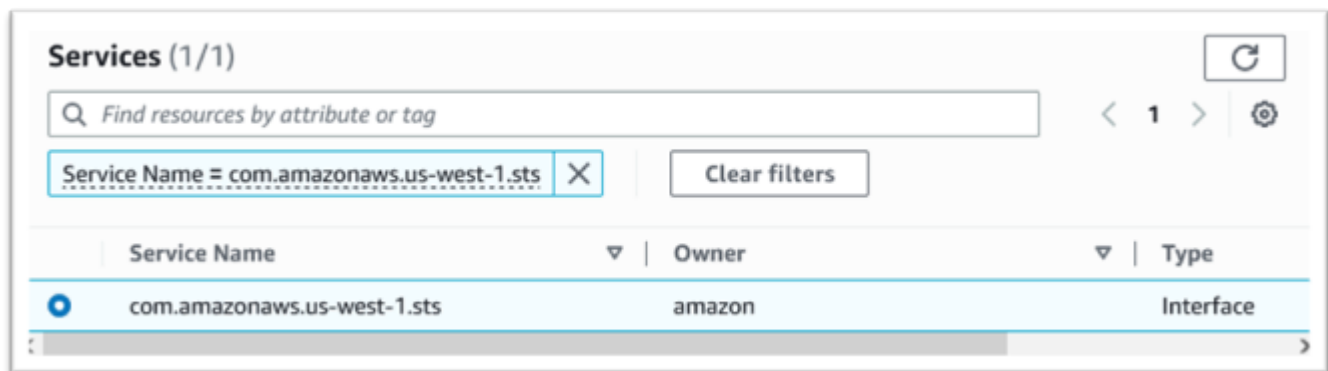
com.amazonaws. *regione* .ec2:



The screenshot shows the AWS IAM console 'Services' page with 2 items. A filter is applied: 'com.amazonaws.us-west-1.ec2'. The table below shows the results:

Service Name	Owner	Type
com.amazonaws.us-west-1.ec2	amazon	Interface
com.amazonaws.us-west-1.ec2messages	amazon	Interface

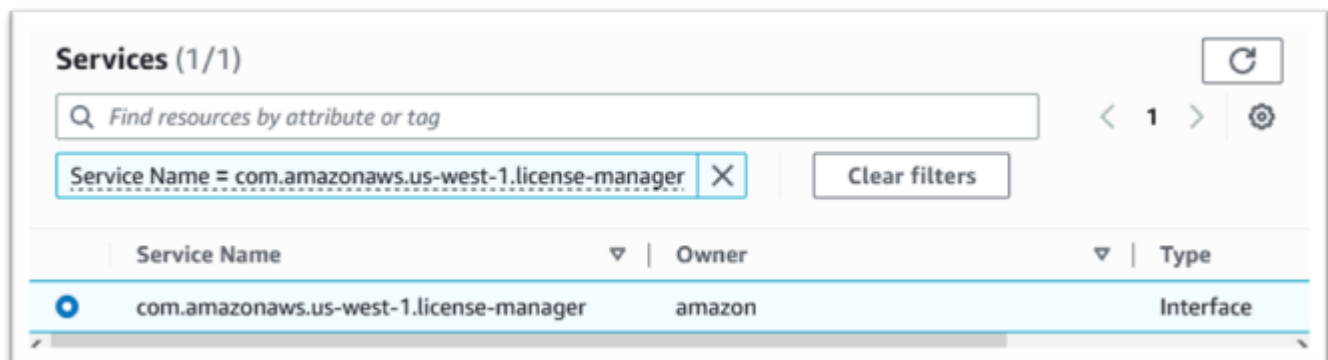
com.amazonaws. **regione** .sts:



The screenshot shows the AWS IAM console 'Services' page with 1 item. A filter is applied: 'Service Name = com.amazonaws.us-west-1.sts'. The table below shows the results:

Service Name	Owner	Type
com.amazonaws.us-west-1.sts	amazon	Interface

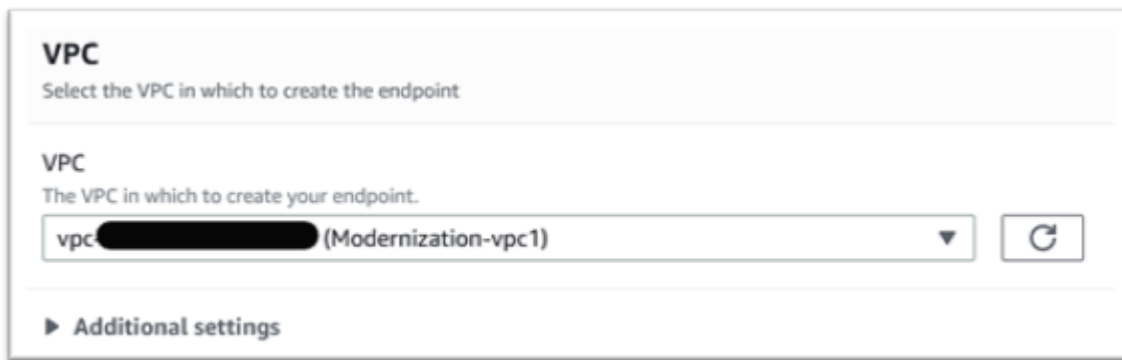
com.amazonaws. **regione** .license-manager:



The screenshot shows the AWS IAM console 'Services' page with 1 item. A filter is applied: 'Service Name = com.amazonaws.us-west-1.license-manager'. The table below shows the results:

Service Name	Owner	Type
com.amazonaws.us-west-1.license-manager	amazon	Interface

7. Per VPC scegli il VPC per l'istanza.



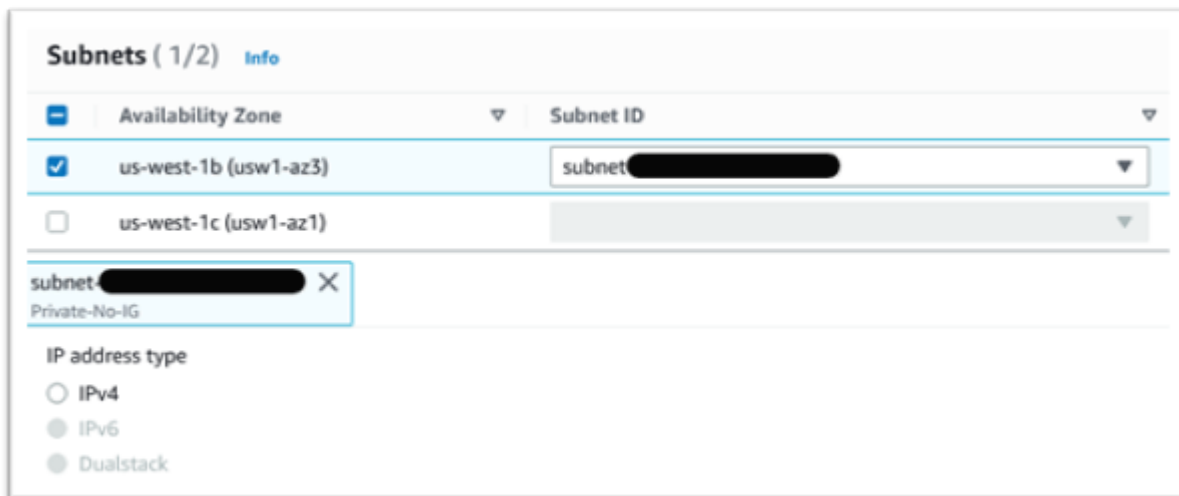
**VPC**  
Select the VPC in which to create the endpoint

**VPC**  
The VPC in which to create your endpoint.

vpc-██████████ (Modernization-vpc1) [Refresh]

▶ Additional settings

8. Scegli la zona di disponibilità e le sottoreti per il VPC.



**Subnets ( 1/2 )** Info

<input type="checkbox"/>	Availability Zone	Subnet ID
<input checked="" type="checkbox"/>	us-west-1b (usw1-az3)	subnet-██████████
<input type="checkbox"/>	us-west-1c (usw1-az1)	██████████

subnet-██████████ X  
Private-No-IG

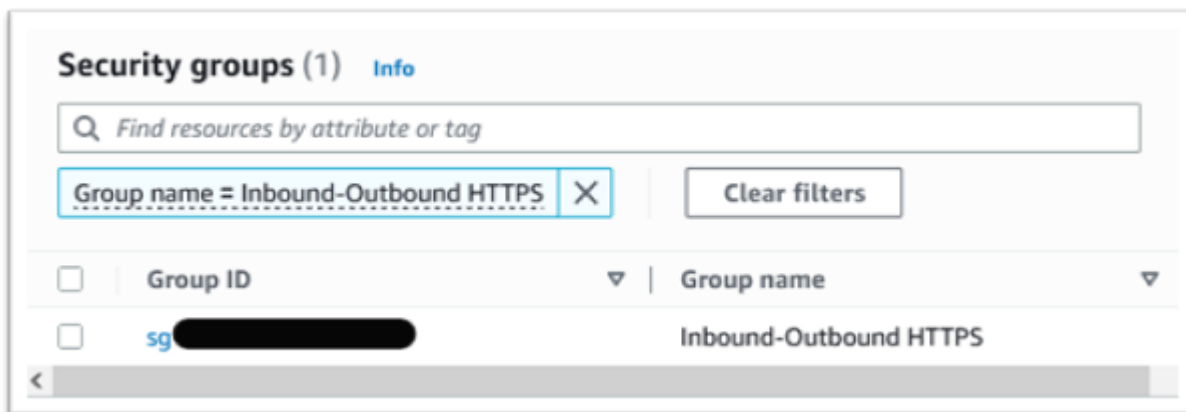
IP address type

IPv4

IPv6

Dualstack

9. Scegli il gruppo di sicurezza creato in precedenza.



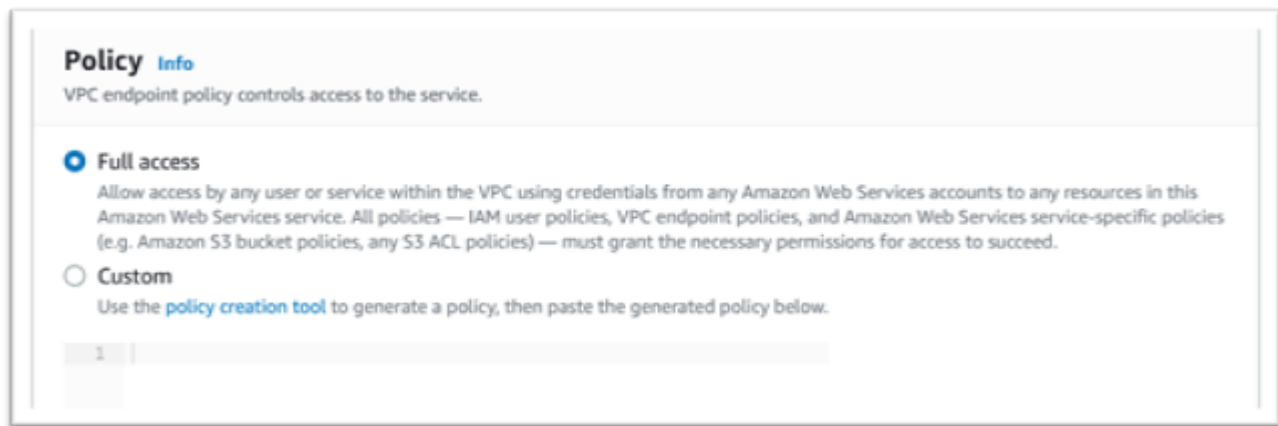
**Security groups (1)** Info

Find resources by attribute or tag

Group name = Inbound-Outbound HTTPS X Clear filters

<input type="checkbox"/>	Group ID	Group name
<input type="checkbox"/>	sg-██████████	Inbound-Outbound HTTPS

10. In Policy scegli Accesso completo.



11. Scegliere Create Endpoint (Crea endpoint).
12. Ripetere questa procedura per le interfacce rimanenti.

## Risoluzione dei problemi relativi alla licenza

In caso di problemi di accesso o utilizzo delle AMI, le seguenti informazioni potrebbero esserti utili.

### Verifica che l'istanza Amazon EC2 abbia il ruolo di licenza IAM

Questo può essere verificato nella scheda Sicurezza dei dettagli dell'istanza Amazon EC2. Questo può essere modificato utilizzando l'opzione di sicurezza del menu a discesa Azioni.

The screenshot shows the AWS Management Console interface for an Amazon EC2 instance. At the top, there is a search bar and filter options. The filters are set to 'Instance state = running' and 'Name = Enterprise Analyzer'. Below the filters, a table lists the instance details:

Name	Instance ID	Instance state	Instance type
Enterprise Analyzer	i-██████████	Running	r6i.xlarge

Below the table, the instance details for 'Enterprise Analyzer' are shown. The 'Security' tab is selected, displaying the following information:

- IAM Role:** Micro-Focus-Licensing-role
- Owner ID:** ██████████
- Security groups:** sg-██████████ (M2-Managed)

## Usa il Reachability Analyzer

Trova il Reachability Analyzer nella pagina AWS Network Manager Console.

Crea e analizza un percorso tra l'istanza Amazon EC2 creata dall'AMI e l'endpoint VPC Amazon S3.

Se l'istanza Amazon EC2 non dispone di accesso a Internet, ripeti l'analisi del percorso su tutti e 4 gli endpoint.

Per ulteriori informazioni sul Reachability Analyzer, [vedere Guida introduttiva a Reachability Analyzer nella guida Reachability Analyzer](#).

## Esegui il license-daemon

In Windows Enterprise Developer usa il seguente comando da un prompt dei comandi:

```
"C:\Program Files (x86)\Micro Focus\Enterprise Developer\AdoptOpenJDK\bin\java" -jar
"C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

ed esaminate l'output. Ignora i messaggi SLF4J e cerca la prima eccezione.

In Enterprise Analyzer utilizzate il seguente comando da un prompt dei comandi:

```
"C:\Program Files (x86)\Micro Focus\AdoptOpenJDK\bin\java" -jar "C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

ed esaminate l'output. Ignora i messaggi SLF4J e cerca la prima eccezione.

Su Linux esegui:

```
java -jar /var/microfocuslicensing/bin/aws-license-daemon.jar
```

Ignora i messaggi SLF4J e cerca la prima eccezione.

Ad esempio, se la risorsa Amazon S3 non è disponibile, l'eccezione è la seguente:

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.

Exception in thread "main" software.amazon.awssdk.services.s3.model.S3Exception: Access
Denied (Service: S3, Status Code: 403, Request ID: P6
```

Il messaggio di eccezione indica quale risorsa non è disponibile. Confrontate i valori di configurazione con quelli mostrati in questo argomento.

## Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer

AWS La modernizzazione del mainframe fornisce diversi strumenti tramite Amazon AppStream 2.0. AppStream 2.0 è un servizio di streaming di applicazioni sicuro e completamente gestito che consente di trasmettere in streaming le applicazioni desktop agli utenti senza riscrivere le applicazioni. AppStream 2.0 offre agli utenti l'accesso istantaneo alle applicazioni di cui hanno bisogno con un'esperienza utente reattiva e fluida sul dispositivo di loro scelta. L'utilizzo della AppStream versione 2.0 per ospitare strumenti specifici del motore di esecuzione offre ai team applicativi dei clienti la possibilità di utilizzare gli strumenti direttamente dai propri browser Web, interagendo con i file delle applicazioni archiviati nei bucket o nei repository di Amazon S3. CodeCommit

Per informazioni sul supporto dei browser nella AppStream versione 2.0, consulta [Requisiti di sistema e supporto delle funzionalità \(browser Web\)](#) nella Guida all'amministrazione di Amazon AppStream 2.0. In caso di problemi con l'utilizzo della AppStream versione 2.0, consulta [Risoluzione dei problemi degli utenti AppStream 2.0](#) nella Guida all'amministrazione di Amazon AppStream 2.0.

Questo documento è destinato ai membri del team operativo dei clienti. Descrive come configurare flotte e stack Amazon AppStream 2.0 per ospitare gli strumenti Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer utilizzati con AWS Mainframe Modernization. Micro Focus Enterprise Analyzer viene solitamente utilizzato durante la fase di valutazione e Micro Focus Enterprise Developer viene solitamente utilizzato durante la fase di migrazione e modernizzazione dell'approccio di modernizzazione del AWS mainframe. Se prevedi di utilizzare sia Enterprise Analyzer che Enterprise Developer, devi creare flotte e stack separati per ogni strumento. Ogni strumento richiede la propria flotta e il proprio stack perché i termini di licenza sono diversi.

#### Important

I passaggi di questo tutorial si basano sul AWS CloudFormation modello scaricabile [cfn-m2](#) - .yaml. `appstream-fleet-ea-ed`

## Argomenti

- [Prerequisiti](#)
- [Passaggio 1: ottieni le immagini AppStream 2.0](#)
- [Passaggio 2: crea lo stack utilizzando il modello AWS CloudFormation](#)
- [Fase 3: Creare un utente nella AppStream versione 2.0](#)
- [Fase 4: Accedere alla AppStream versione 2.0](#)
- [Passaggio 5: verifica i bucket in Amazon S3 \(opzionale\)](#)
- [Fasi successive](#)
- [Pulizia delle risorse](#)

## Prerequisiti

- Scarica il modello: [cfn-m2- appstream-fleet-ea-ed](#) .yaml.
- Ottieni l'ID del tuo VPC e del tuo gruppo di sicurezza predefiniti. Per ulteriori informazioni sul cloud privato virtuale predefinito, consulta la [Guida](#) per l'utente di Amazon VPC. Per ulteriori informazioni



sul gruppo di sicurezza predefinito, consulta [Gruppi di sicurezza predefiniti e personalizzati](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

- Assicurati di disporre delle seguenti autorizzazioni:
  - crea pile, flotte e utenti in AppStream 2.0.
  - crea pile AWS CloudFormation utilizzando un modello.
  - crea bucket e carica file nei bucket in Amazon S3.
  - scarica le credenziali (access\_key\_idsecret\_access\_key) da IAM.

## Passaggio 1: ottieni le immagini AppStream 2.0

In questo passaggio, condividi le immagini AppStream 2.0 per Enterprise Analyzer ed Enterprise Developer con il tuo AWS account.

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nella barra di navigazione a sinistra, scegli Strumenti.
3. In Analisi, sviluppo e creazione di risorse, scegli Condividi risorse con il mio AWS account.

## Passaggio 2: crea lo stack utilizzando il modello AWS CloudFormation

In questa fase, utilizzate il AWS CloudFormation modello scaricato per creare uno stack AppStream 2.0 e un parco veicoli per l'esecuzione di Micro Focus Enterprise Analyzer. È possibile ripetere questo passaggio in seguito per creare un altro stack e un altro parco veicoli AppStream 2.0 per l'esecuzione di Micro Focus Enterprise Developer, poiché ogni strumento richiede il proprio parco macchine e stack in AppStream 2.0. Per ulteriori informazioni sugli AWS CloudFormation stack, consulta [Lavorare con gli stack nella Guida](#) per l'AWS CloudFormationutente.

### Note

AWSMainframe Modernization aggiunge un costo aggiuntivo ai prezzi standard AppStream 2.0 per l'utilizzo di Enterprise Analyzer ed Enterprise Developer. Per ulteriori informazioni, consulta la pagina relativa ai prezzi della [modernizzazione dei AWS mainframe](#).

1. Scarica il modello [cfn-m2- appstream-fleet-ea-ed .yml](#), se necessario.

2. Apri la AWS CloudFormation console e scegli Crea stack e con nuove risorse (standard).
3. In Prerequisiti - Prepara modello, scegli Il modello è pronto.
4. In Specifica modello, scegli Carica un file modello.
5. In Carica un file modello, scegli il file e carica il modello [cfn-m2- appstream-fleet-ea-ed .yaml](#).
6. Seleziona Successivo.

The screenshot shows the 'Create stack' wizard in the AWS CloudFormation console. The current step is 'Specify template'. The interface includes a sidebar with steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The main content area is divided into two sections: 'Prerequisite - Prepare template' and 'Specify template'. In the 'Prerequisite' section, the 'Template is ready' radio button is selected. In the 'Specify template' section, the 'Upload a template file' radio button is selected. Below this, the 'Choose file' button is highlighted, and the filename 'cfn-m2-appstream-fleet-ea-ed.yaml' is visible. A 'View in Designer' button is also present. At the bottom right, there are 'Cancel' and 'Next' buttons.

7. In Specifica i dettagli dello stack, inserisci le seguenti informazioni:
  - In Nome dello stack, inserisci un nome a tua scelta. Ad esempio, **m2-ea**.
  - Nel AppStreamApplication, scegli tè.
  - In AppStreamFleetSecurityGroup, scegli il gruppo di sicurezza predefinito del tuo VPC predefinito.
  - Nel AppStreamFleetVpcSubnet, scegli una sottorete all'interno del tuo VPC predefinito.
  - In AppStreamImageName, scegli l'immagine che inizia con `m2-enterprise-analyzer`. Questa immagine contiene la versione attualmente supportata dello strumento Micro Focus Enterprise Analyzer.
  - Accetta le impostazioni predefinite per gli altri campi, quindi scegli Avanti.

Step 1  
**Specify template**


Step 2  
**Specify stack details**

Step 3  
Configure stack options


Step 4  
Review


## Specify stack details

**Stack name**


Stack name   
m2-ea-2  
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).


**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**AppStreamApplication**   
AppStream application  
ea

**AppStreamFleetSecurityGroup**   
AppStream fleet security group  
sg-27c2fb57

**AppStreamFleetType**  
AppStream fleet type  
ALWAYS\_ON

**AppStreamFleetVpcSubnet**   
AppStream fleet subnet  
subnet-57f8a30d

**AppStreamImageName**   
AppStream machine image name: m2-enterprise-analyzer-v7.0.1.R1 or m2-enterprise-developer-v7.0.3.R1  
m2-enterprise-analyzer-v7.0.1.R1

**AppStreamInstanceType**  
AppStream instance type  
stream.standard.large

**AppStreamInstances**  
AppStream desired instances  
2

**AppStreamView**  
AppStream view  
DESKTOP

Cancel Previous **Next**

- Accetta tutte le impostazioni predefinite, quindi scegli nuovamente Avanti.
- In fase di revisione, assicurati che tutti i parametri siano quelli che intendi.
- Scorri fino alla fine, scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati e scegli Create Stack.

La creazione dello stack e della flotta richiede dai 20 ai 30 minuti. Puoi scegliere Aggiorna per vedere gli AWS CloudFormation eventi man mano che si verificano.

## Fase 3: Creare un utente nella AppStream versione 2.0

Mentre aspetti di AWS CloudFormation completare la creazione dello stack, puoi creare uno o più utenti nella AppStream versione 2.0. Questi utenti sono coloro che utilizzeranno Enterprise Analyzer nella AppStream versione 2.0. Dovrai specificare un indirizzo e-mail per ogni utente e assicurarti che ogni utente disponga delle autorizzazioni sufficienti per creare bucket in Amazon S3, caricare file in un bucket e collegarsi a un bucket per mapparne i contenuti.

1. Apri la console AppStream 2.0.
2. Nella barra di navigazione a sinistra, scegli Pool di utenti.
3. Selezionare Create user (Crea utente).
4. Fornisci un indirizzo e-mail in cui l'utente possa ricevere un invito via e-mail all'uso AppStream 2.0, un nome e cognome e scegli Crea utente.
5. Ripetere l'operazione se necessario per creare altri utenti. L'indirizzo e-mail di ogni utente deve essere univoco.

Per ulteriori informazioni sulla creazione di utenti AppStream 2.0, consulta i [pool di utenti AppStream 2.0](#) nella Amazon AppStream 2.0 Administration Guide.

Al AWS CloudFormation termine della creazione dello stack, puoi assegnare l'utente che hai creato allo stack, come segue:

1. Apri la console AppStream 2.0.
2. Scegli il nome utente.
3. Scegli Azione, quindi Assegna pila.
4. In Assegna pila, scegli la pila che inizia con. `m2-appstream-stack-ea`
5. Scegliere Assign stack (Assegna stack).

**Assign stack** ✕

Select a stack to enable access to the user(s) below.

User(s) being assigned

- Mary Major (mary.major@example.com)

Stack

m2-appstream-stack-ea-c92d75b0 ▼

Send email notification to user

Cancel Assign stack

L'assegnazione di un utente a uno stack fa sì che AppStream 2.0 invii un'e-mail all'utente all'indirizzo fornito. Questa e-mail contiene un link alla pagina di accesso AppStream 2.0.

## Fase 4: Accedere alla AppStream versione 2.0

In questo passaggio, accedi alla AppStream versione 2.0 utilizzando il link contenuto nell'e-mail inviata da AppStream 2.0 all'utente che hai creato [Fase 3: Creare un utente nella AppStream versione 2.0](#).

1. Accedi alla AppStream versione 2.0 utilizzando il link fornito nell'e-mail inviata da AppStream 2.0.
2. Cambia la password, se richiesto. La schermata AppStream 2.0 che vedi è simile alla seguente:



3. Scegli Desktop.
4. Nella barra delle applicazioni, scegli Cerca e inserisci **D**: per accedere alla cartella Home.

#### Note

Se salti questo passaggio, potresti ricevere un `Device not ready` errore quando provi ad accedere alla cartella Home.

In qualsiasi momento, se riscontri problemi con l'accesso alla AppStream versione 2.0, puoi riavviare il tuo parco veicoli AppStream 2.0 e provare ad accedere nuovamente, seguendo i seguenti passaggi.

1. Apri la console AppStream 2.0.
2. Nella barra di navigazione a sinistra, scegli Flotte.
3. Scegli la flotta che stai cercando di utilizzare.
4. Scegli Azione, quindi scegli Stop.
5. Aspetta che la flotta si fermi.
6. Scegli Azione, quindi scegli Avvia.

Questo processo può richiedere circa 10 minuti.

## Passaggio 5: verifica i bucket in Amazon S3 (opzionale)

Una delle attività completate dal AWS CloudFormation modello utilizzato per creare lo stack è stata la creazione di due bucket in Amazon S3, necessari per salvare e ripristinare i dati utente e le impostazioni delle applicazioni durante le sessioni di lavoro. Questi secchi sono i seguenti:

- Il nome inizia con `onappstream2-`. Questo bucket mappa i dati nella tua cartella Home in AppStream 2.0 (D:\PhotonUser\My Files\Home Folder).

### Note

La cartella Home è unica per un determinato indirizzo e-mail ed è condivisa tra tutte le flotte e gli stack di un determinato AWS account. Il nome della cartella Home è un hash SHA256 dell'indirizzo e-mail dell'utente ed è memorizzato su un percorso basato su tale hash.

- Il nome inizia con `onappstream-app-settings-`. Questo bucket contiene informazioni sulla sessione utente per la AppStream versione 2.0 e include impostazioni come i preferiti del browser, i profili di connessione IDE e delle applicazioni e le personalizzazioni dell'interfaccia utente. Per ulteriori informazioni, consulta [How Application Settings Persistence Works](#) nella Amazon AppStream 2.0 Administration Guide.

Per verificare che i bucket siano stati creati, segui questi passaggi:

1. Aprire la console Amazon S3.
2. Nella barra di navigazione a sinistra, scegli Bucket.
3. In Trova bucket per nome, inserisci **appstream** per filtrare l'elenco.

Se vedi i bucket, non sono necessarie ulteriori azioni. Tieni presente che i secchi esistono. Se non vedi i bucket, significa che il AWS CloudFormation modello non è ancora stato eseguito o si è verificato un errore. Vai alla AWS CloudFormation console e rivedi i messaggi di creazione dello stack.

## Fasi successive

Ora che l'infrastruttura AppStream 2.0 è configurata, è possibile configurare e iniziare a utilizzare Enterprise Analyzer. Per ulteriori informazioni, consulta [Tutorial: configurare Enterprise Analyzer su](#)

[2.0 AppStream](#) . Puoi anche configurare Enterprise Developer. Per ulteriori informazioni, consulta [Tutorial: configurare Micro Focus Enterprise Developer sulla AppStream versione 2.0](#).

## Pulizia delle risorse

La procedura per ripulire lo stack e le flotte create è descritta in [Create an Fleet and Stack AppStream 2.0](#).

Una volta eliminati gli oggetti AppStream 2.0, l'amministratore dell'account può anche, se appropriato, ripulire i bucket Amazon S3 per le impostazioni dell'applicazione e le cartelle home.

### Note

La cartella home di un determinato utente è unica per tutte le flotte, quindi potrebbe essere necessario conservarla se nello stesso account sono attivi altri stack AppStream 2.0.

Infine, la AppStream versione 2.0 attualmente non consente di eliminare utenti tramite la console. È invece necessario utilizzare l'API del servizio con la CLI. Per ulteriori informazioni, consulta [Amministrazione del pool di utenti](#) nella Guida all'amministrazione di Amazon AppStream 2.0.

## Tutorial: configurare Enterprise Analyzer su 2.0 AppStream

Questo tutorial descrive come configurare Micro Focus Enterprise Analyzer per analizzare una o più applicazioni mainframe. Lo strumento Enterprise Analyzer fornisce diversi report basati sull'analisi del codice sorgente dell'applicazione e delle definizioni di sistema.

Questa configurazione è progettata per favorire la collaborazione in team. L'installazione utilizza un bucket Amazon S3 per condividere il codice sorgente con dischi virtuali. In questo modo si utilizza [Rclone](#)) sul computer Windows. Con un'istanza Amazon RDS comune che esegue [PostgreSQL](#), qualsiasi membro del team può accedere a tutti i report richiesti.

I membri del team possono anche montare il disco virtuale con backup di Amazon S3 sui propri computer personali e aggiornare il bucket di origine dalle proprie workstation. Possono potenzialmente utilizzare script o qualsiasi altra forma di automazione sui propri computer se sono collegati ad altri sistemi interni locali.

La configurazione si basa sulle immagini Windows AppStream 2.0 che AWS Mainframe Modernization condivide con il cliente. La configurazione si basa inoltre sulla creazione di flotte e



stack AppStream 2.0, come descritto in. [Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#)

### Important

I passaggi di questo tutorial presuppongono che la AppStream versione 2.0 venga configurata con il AWS CloudFormation modello scaricabile `cfn-m2- .yaml. appstream-fleet-ea-ed` Per ulteriori informazioni, consulta [Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#).

Per eseguire i passaggi descritti in questo tutorial, è necessario aver configurato la flotta e lo stack di Enterprise Analyzer e devono essere in esecuzione.

Per una descrizione completa delle funzionalità e dei risultati di Enterprise Analyzer, consultate la [documentazione di Enterprise Analyzer sul sito Web Micro Focus](#).

## Contenuto dell'immagine

Oltre all'applicazione Enterprise Analyzer stessa, l'immagine contiene i seguenti strumenti e librerie.

Strumenti di terze parti

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [gitscm](#)
- [Driver ODBC PostgreSQL](#)

Librerie in `C:\Users\Public`

- BankDemo codice sorgente e definizione del progetto per Enterprise Developer: `m2-bankdemo-template.zip`.
- Pacchetto di installazione MFA per il mainframe: `mfa.zip` Per ulteriori informazioni, vedere [Mainframe Access Overview](#) nella documentazione di Micro Focus Enterprise Developer.
- File di comando e configurazione per Rclone (istruzioni per il loro utilizzo nei tutorial): `e. m2-rclone.cmd m2-rclone.conf`

## Argomenti

- [Prerequisiti](#)
- [Fase 1: configurazione](#)
- [Passaggio 2: creare la cartella virtuale basata su Amazon S3 su Windows](#)
- [Fase 3: creare una fonte ODBC per l'istanza Amazon RDS](#)
- [Sessioni successive](#)
- [Risoluzione dei problemi di connessione all'area di lavoro](#)
- [Pulizia delle risorse](#)

## Prerequisiti

- Carica il codice sorgente e le definizioni di sistema per l'applicazione del cliente che desideri analizzare in un bucket S3. Le definizioni di sistema includono CICS CSD, definizioni di oggetti DB2 e così via. È possibile creare una struttura di cartelle all'interno del bucket adatta al modo in cui si desidera organizzare gli elementi dell'applicazione. Ad esempio, quando decomprimete l' `BankDemo` esempio, ha la seguente struttura:

```
demo
  |--> jcl
  |--> RDEF
  |--> transaction
  |--> xa
```

- Crea e avvia un'istanza Amazon RDS che esegue PostgreSQL. Questa istanza memorizzerà i dati e i risultati prodotti da Enterprise Analyzer. È possibile condividere questa istanza con tutti i membri del team dell'applicazione. Inoltre, create uno schema vuoto chiamato `m2_ea` (o qualsiasi altro nome adatto) nel database. Definisci le credenziali per gli utenti autorizzati che consentono loro di creare, inserire, aggiornare ed eliminare elementi in questo schema. Puoi ottenere il nome del database, l'URL dell'endpoint del server e la porta TCP dalla console Amazon RDS o dall'amministratore dell'account.
- Assicurati di aver configurato l'accesso programmatico al tuo Account AWS. Per ulteriori informazioni, vedere [Accesso programmatico](#) in Riferimenti generali di Amazon Web Services

## Fase 1: configurazione

1. Avviate una sessione con la AppStream versione 2.0 con l'URL che avete ricevuto nel messaggio e-mail di benvenuto della AppStream versione 2.0.
2. Usa la tua e-mail come ID utente e definisci la tua password permanente.
3. Seleziona lo stack Enterprise Analyzer.
4. Nella pagina del menu AppStream 2.0, scegli Desktop per accedere al desktop Windows su cui è in streaming la flotta.

## Passaggio 2: creare la cartella virtuale basata su Amazon S3 su Windows

### Note

Se hai già utilizzato Rclone durante l'anteprima di AWS Mainframe Modernization, devi eseguire l'aggiornamento `m2-rclone.cmd` alla versione più recente disponibile in: `C:\Users\Public`

1. Copia i `m2-rclone.cmd` file `m2-rclone.conf` and forniti nella tua cartella home usando File Explorer. `C:\Users\PhotonUser\My Files\Home Folder`
2. Aggiorna i parametri di `m2-rclone.conf` configurazione con la tua chiave di AWS accesso e il segreto corrispondente, oltre al tuo Regione AWS.

```
[m2-s3]
type = s3
provider = AWS
access_key_id = YOUR-ACCESS-KEY
secret_access_key = YOUR-SECRET-KEY
region = YOUR-REGION
acl = private
server_side_encryption = AES256
```

3. In `m2-rclone.cmd`, apporta le seguenti modifiche:
  - Cambia `your-s3-bucket` il nome del tuo bucket Amazon S3. Ad esempio, `m2-s3-mybucket`.
  - Passa `your-s3-folder-key` alla tua bucket key Amazon S3. Ad esempio, `myProject`.

- Passa `your-local-folder-path` al percorso della directory in cui desideri sincronizzare i file dell'applicazione dal bucket Amazon S3 che li contiene. Ad esempio, `D:\PhotonUser\My Files\Home Folder\m2-new`. Questa directory sincronizzata deve essere una sottodirectory della cartella Home per consentire alla AppStream versione 2.0 di eseguire correttamente il backup e il ripristino all'inizio e alla fine della sessione.

```
:loop
timeout /T 10
"C:\Program Files\rclone\rclone.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rclone.conf"
goto :loop
```

4. Apri un prompt dei comandi di Windows, cd to `C:\Users\PhotonUser\My Files\Home Folder` se necessario ed esegui `m2-rclone.cmd`. Questo script di comandi esegue un ciclo continuo, sincronizzando il bucket e la chiave Amazon S3 con la cartella locale ogni 10 secondi. Puoi regolare il timeout in base alle tue esigenze. Dovresti vedere il codice sorgente dell'applicazione che si trova nel bucket Amazon S3 in Windows File Explorer.

Per aggiungere nuovi file al set su cui stai lavorando o aggiornare quelli esistenti, carica i file nel bucket Amazon S3 e verranno sincronizzati con la tua directory alla successiva iterazione definita in `m2-rclone.cmd`. Allo stesso modo, se desideri eliminare alcuni file, eliminali dal bucket Amazon S3. La prossima operazione di sincronizzazione li eliminerà dalla directory locale.

## Fase 3: creare una fonte ODBC per l'istanza Amazon RDS

1. Per avviare lo strumento EA\_Admin, accedi al menu di selezione delle applicazioni nell'angolo superiore sinistro della finestra del browser e scegli MF EA\_Admin.
2. Dal menu Amministra, scegliete Origini dati ODBC e scegliete Aggiungi dalla scheda DSN utente.
3. Nella finestra di dialogo Crea nuova origine dati, scegli il driver Unicode PostgreSQL, quindi scegli Fine.
4. Nella finestra di dialogo Configurazione PostgreSQL Unicode ODBC Driver (pSQLOdbc), definisci e prendi nota del nome dell'origine dati che desideri. Completa i seguenti parametri con i valori dell'istanza RDS creata in precedenza:

## Descrizione

Descrizione opzionale per aiutarti a identificare rapidamente questa connessione al database.

## Database

Il database Amazon RDS che hai creato in precedenza.

## Server

L'endpoint Amazon RDS.

## Porta

La porta Amazon RDS.

## Nome utente

Come definito nell'istanza Amazon RDS.

## Password

Come definito nell'istanza Amazon RDS.

5. Scegli Test per verificare che la connessione ad Amazon RDS sia riuscita, quindi scegli Salva per salvare il tuo nuovo DSN utente.
6. Attendi di visualizzare il messaggio che conferma la creazione dell'area di lavoro corretta, quindi scegli OK per terminare con ODBC Data Sources e chiudere lo strumento EA\_Admin.
7. Passate nuovamente al menu di selezione dell'applicazione e scegliete Enterprise Analyzer per avviare lo strumento. Scegliete Crea nuovo.
8. Nella finestra di configurazione dell'area di lavoro, inserisci il nome dell'area di lavoro e definisci la sua posizione. L'area di lavoro può essere il disco basato su Amazon S3 se utilizzi questa configurazione o la cartella home se preferisci.
9. Scegli Scegli altro database per connetterti alla tua istanza Amazon RDS.
10. Scegli l'icona Postgre tra le opzioni, quindi scegli OK.
11. Per le impostazioni di Windows in Opzioni — Definisci parametri di connessione, inserisci il nome della fonte di dati che hai creato. Immettete anche il nome del database, il nome dello schema, il nome utente e la password. Scegli OK.
12. Attendi che Enterprise Analyzer crei tutte le tabelle, gli indici e così via per memorizzare i risultati. Questo processo potrebbe richiedere un paio di minuti. Enterprise Analyzer conferma quando il database e l'area di lavoro sono pronti per l'uso.

13. Passate nuovamente al menu di selezione dell'applicazione e scegliete Enterprise Analyzer per avviare lo strumento.
14. La finestra di avvio di Enterprise Analyzer viene visualizzata nella nuova posizione selezionata dell'area di lavoro. Scegli OK.
15. Accedete al repository nel riquadro di sinistra, selezionate il nome del repository e scegliete Aggiungi file/cartelle allo spazio di lavoro. Selezionate la cartella in cui è memorizzato il codice dell'applicazione per aggiungerlo all'area di lavoro. Se lo desideri, puoi utilizzare il codice di BankDemo esempio precedente. Quando Enterprise Analyzer richiede di verificare tali file, scegliete Verifica per avviare il rapporto di verifica iniziale di Enterprise Analyzer. Il completamento del processo potrebbe richiedere alcuni minuti, a seconda delle dimensioni dell'applicazione.
16. Espandi l'area di lavoro per visualizzare i file e le cartelle che hai aggiunto all'area di lavoro. I tipi di oggetti e i report sulla complessità ciclomatica sono visibili anche nel quadrante superiore del riquadro Chart Viewer.

È ora possibile utilizzare Enterprise Analyzer per tutte le attività necessarie.

## Sessioni successive

1. Avviate una sessione con la AppStream versione 2.0 con l'URL che avete ricevuto nel messaggio e-mail di benvenuto della AppStream versione 2.0.
2. Accedi con la tua e-mail e la password permanente.
3. Seleziona lo stack Enterprise Analyzer.
4. Avvia Rclone per connetterti al disco supportato da Amazon S3 se utilizzi questa opzione per condividere i file dell'area di lavoro.
5. Avvia Enterprise Analyzer per svolgere le tue attività.

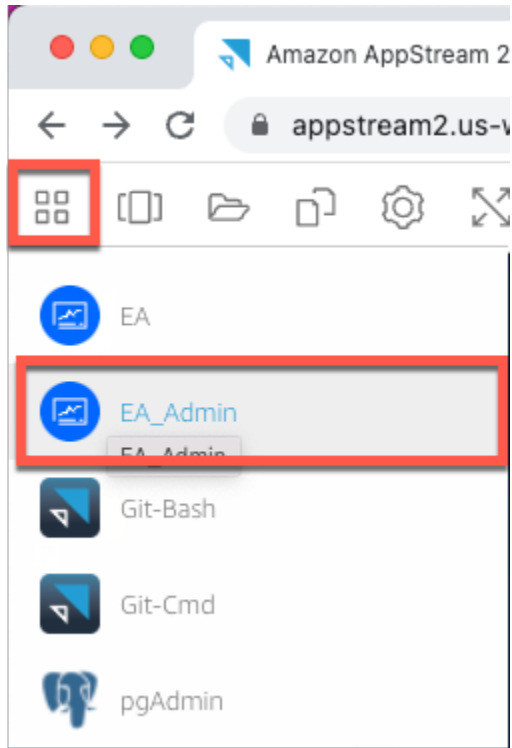
## Risoluzione dei problemi di connessione all'area di lavoro

Quando si tenta di riconnettersi all'area di lavoro di Enterprise Analyzer, è possibile che venga visualizzato un errore simile al seguente:

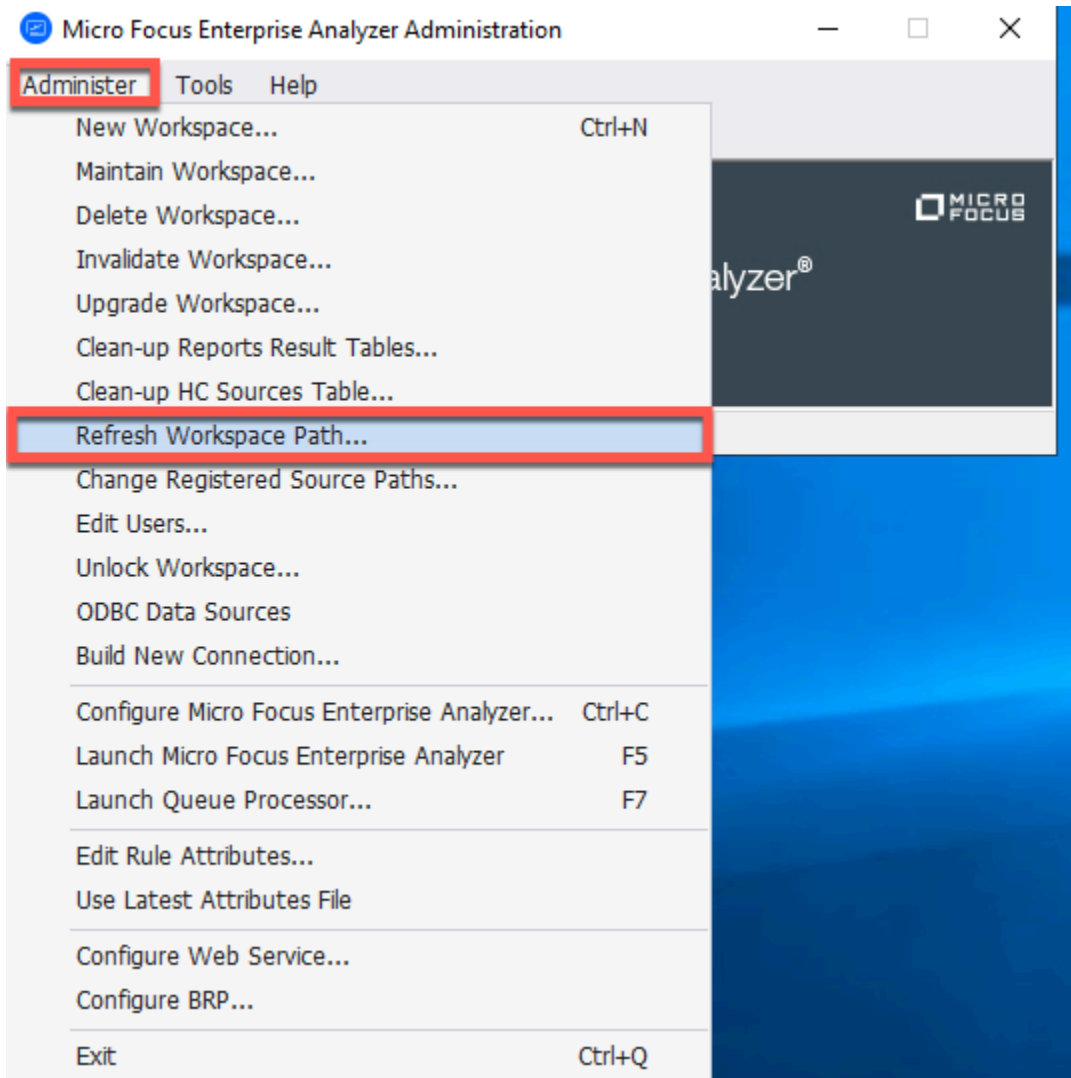
```
Cannot access the workspace directory D:\PhotonUser\My Files\Home Folder\EA_BankDemo.  
The workspace has been created on a non-shared disk of the EC2AMAZ-E6LC33H computer.  
Would you like to correct the workspace directory location?
```

Per risolvere questo problema, scegli OK per cancellare il messaggio, quindi completa i seguenti passaggi.

1. Nella AppStream versione 2.0, scegliete l'icona Avvia applicazione sulla barra degli strumenti, quindi scegliete EA\_Admin per avviare lo strumento di amministrazione di Micro Focus Enterprise Analyzer.

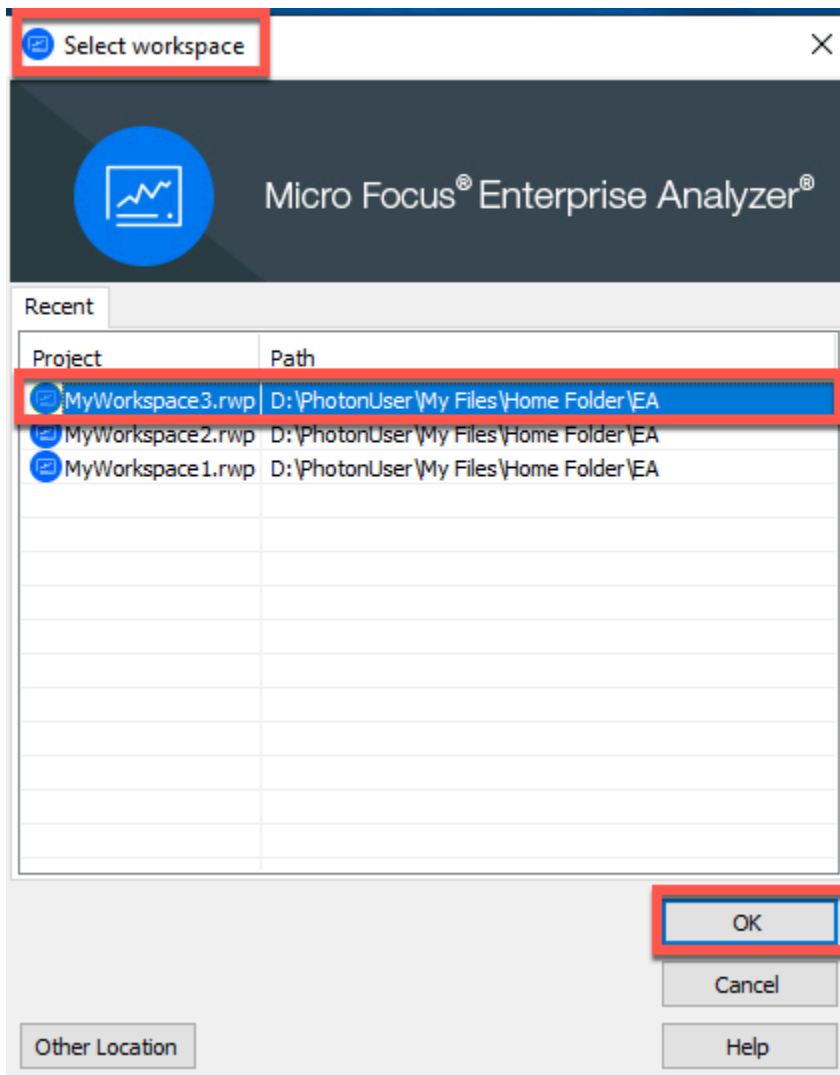


2. Dal menu Amministra, selezionate Aggiorna percorso Workspace... .

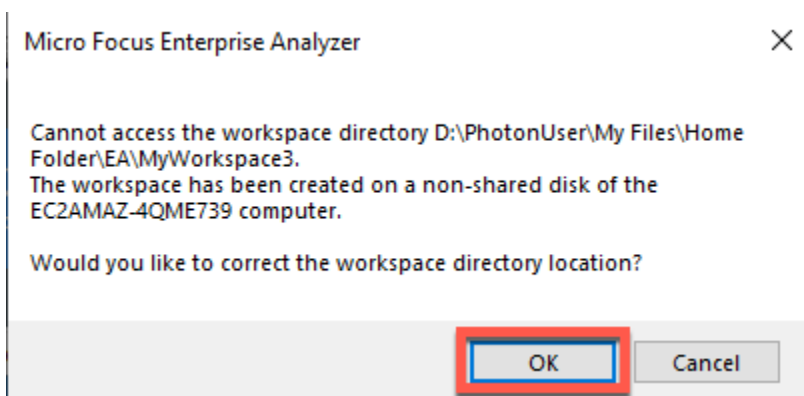


3. In Seleziona area di lavoro, scegliete l'area di lavoro desiderata, quindi fate clic su OK.

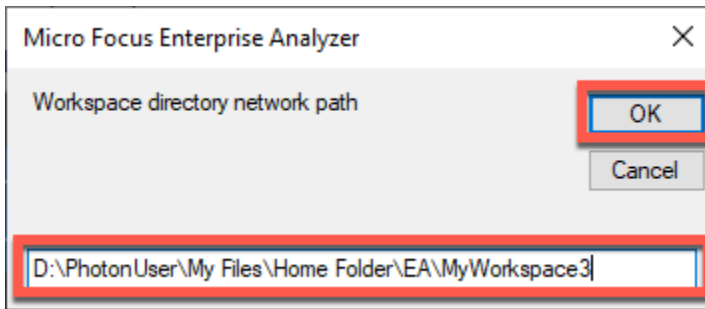




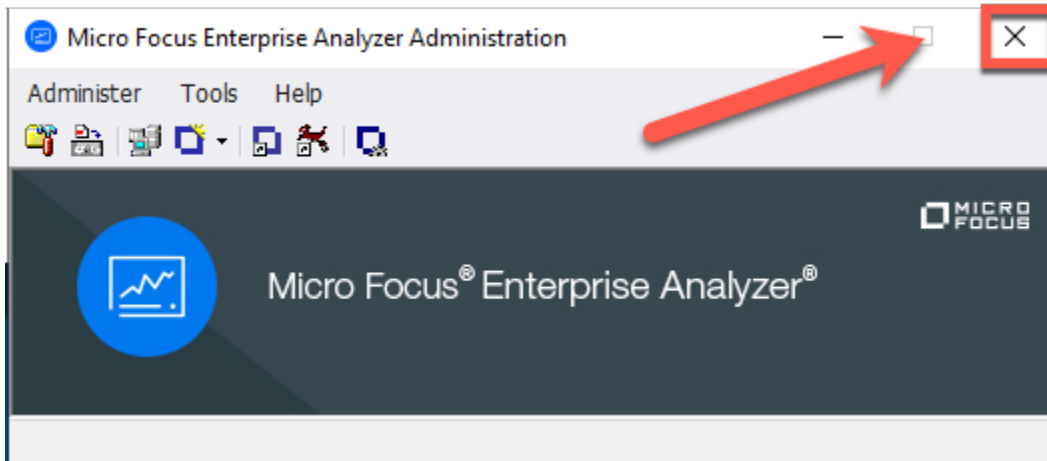
4. Scegliete OK per confermare il messaggio di errore.



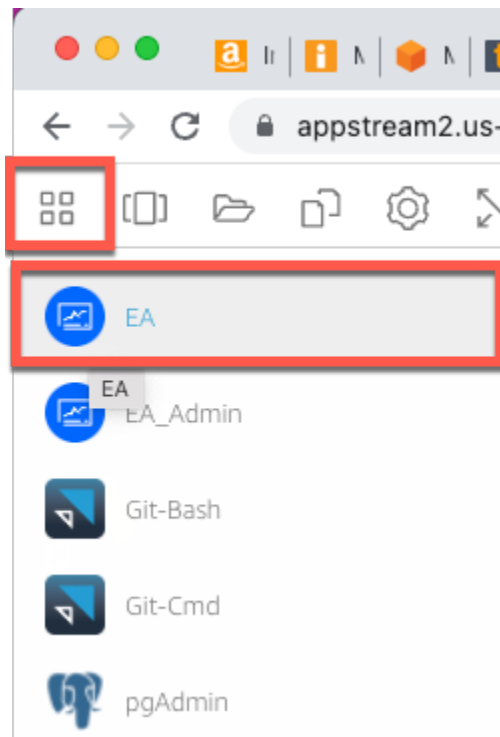
5. In Workspace Directory network path, inserite il percorso corretto del workspace, ad esempio. D:\PhotonUser\My Files\Home Folder\EA\MyWorkspace3



6. Chiudete lo strumento di amministrazione di Micro Focus Enterprise Analyzer.



7. Nella AppStream versione 2.0, scegliete l'icona Avvia applicazione sulla barra degli strumenti, quindi scegliete EA per avviare Micro Focus Enterprise Analyzer.



## 8. Ripetere i passaggi da 3 a 5.

Micro Focus Enterprise Analyzer dovrebbe ora aprirsi con l'area di lavoro esistente.

## Pulizia delle risorse

Se non avete più bisogno delle risorse che avete creato per questo tutorial, eliminatele in modo da non incorrere in ulteriori addebiti. Completa questa procedura:

- Utilizzate lo strumento EA\_Admin per eliminare l'area di lavoro.
- Elimina i bucket S3 che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminazione di un bucket nella Guida](#) per l'utente di Amazon S3.
- Elimina il database che hai creato per questo tutorial. Per ulteriori informazioni, consulta la sezione relativa all'[eliminazione di un'istanza database](#).

## Tutorial: configurare Micro Focus Enterprise Developer sulla AppStream versione 2.0

Questo tutorial descrive come configurare Micro Focus Enterprise Developer per una o più applicazioni mainframe al fine di gestirle, compilarle e testarle utilizzando le funzionalità di Enterprise Developer. La configurazione si basa sulle immagini Windows AppStream 2.0 che AWS Mainframe Modernization condivide con il cliente e sulla creazione di flotte e stack AppStream 2.0 come descritto in [Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#)

### Important

[I passaggi di questo tutorial presuppongono la configurazione della AppStream versione 2.0 utilizzando il modello scaricabile cfn-m2- .yamlAWS CloudFormation. appstream-fleet-ea-ed](#)  
Per ulteriori informazioni, consulta [Tutorial: Configurazione AppStream 2.0 da utilizzare con Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer](#).

È necessario eseguire i passaggi di questa configurazione quando la flotta e lo stack di Enterprise Developer sono attivi e funzionanti.

Per una descrizione completa delle funzionalità e dei risultati di Enterprise Developer v7, consultate la [documentazione up-to-date online \(v7.0\)](#) sul sito Micro Focus.

## Contenuti dell'immagine

Oltre allo stesso Enterprise Developer, l'immagine contiene l'immagine contenente Rumba (un emulatore TN3270). Contiene anche i seguenti strumenti e librerie.

### Strumenti di terze parti

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [gitscm](#)
- [Driver ODBC PostgreSQL](#)

### Librerie in C:\Users\Public

- BankDemo codice sorgente e definizione del progetto per Enterprise Developer:m2-bankdemo-template.zip.
- Pacchetto di installazione MFA per il mainframe:. mfa.zip Per ulteriori informazioni, vedere [Mainframe Access Overview](#) nella documentazione di Micro Focus Enterprise Developer.
- File di comando e configurazione per Rclone (istruzioni per il loro utilizzo nei tutorial): e. m2-rclone.cmd m2-rclone.conf

Se devi accedere al codice sorgente che non è ancora stato caricato nei CodeCommit repository, ma che è disponibile in un bucket Amazon S3, ad esempio per eseguire il caricamento iniziale del codice sorgente in git, segui la procedura per creare un disco Windows virtuale come descritto in. [Tutorial: configurare Enterprise Analyzer su 2.0 AppStream](#)

### Argomenti

- [Prerequisiti](#)
- [Fase 1: Configurazione da parte dei singoli utenti Enterprise Developer](#)
- [Fase 2: creare la cartella virtuale basata su Amazon S3 su Windows \(opzionale\)](#)
- [Passaggio 3: clonare il repository](#)

- [Sessioni successive](#)
- [Pulizia delle risorse](#)

## Prerequisiti

- Uno o più CodeCommit repository caricati con il codice sorgente dell'applicazione da gestire. La configurazione del repository deve soddisfare i requisiti della pipeline CI/CD di cui sopra per creare sinergie combinando entrambi gli strumenti.
- Ogni utente deve disporre delle credenziali per il CodeCommit repository o i repository definiti dall'amministratore dell'account in base alle informazioni in [Authentication and access control for AWS](#). La struttura di tali credenziali è esaminata in [Authentication and access control for AWS CodeCommit](#) e il riferimento completo per le autorizzazioni IAM CodeCommit si trova nel [riferimento alle CodeCommit autorizzazioni](#): l'amministratore può definire politiche IAM distinte per ruoli distinti con credenziali specifiche per il ruolo di ciascun repository e limitando le autorizzazioni dell'utente allo specifico set di attività che deve svolgere su un determinato repository. Pertanto, per ogni manutentore del CodeCommit repository, l'amministratore dell'account genererà un utente principale e concederà a tale utente le autorizzazioni per accedere al repository o ai repository richiesti selezionando la politica o le politiche IAM appropriate per CodeCommit l'accesso.

## Fase 1: Configurazione da parte dei singoli utenti Enterprise Developer

1. Ottieni le tue credenziali IAM:
  1. Connect alla AWS console all'[indirizzo https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/).
  2. Segui la procedura descritta nel passaggio 3 di [Configurazione per gli utenti HTTPS che utilizzano le credenziali Git](#) nella Guida per l'AWS CodeCommit utente.
  3. Copia le credenziali di accesso CodeCommit specifiche che IAM ha generato per te, mostrando, copiando e incollando queste informazioni in un file sicuro sul tuo computer locale oppure scegliendo Scarica credenziali per scaricare queste informazioni come file.CSV. Queste informazioni sono necessarie per connettersi a CodeCommit.
2. Inizia una sessione con AppStream 2.0 in base all'URL ricevuto nell'e-mail di benvenuto. Usa la tua email come nome utente e crea la tua password.
3. Seleziona il tuo stack Enterprise Developer.
4. Nella pagina del menu, scegli Desktop per accedere al desktop di Windows trasmesso in streaming dalla flotta.

## Fase 2: creare la cartella virtuale basata su Amazon S3 su Windows (opzionale)

Se è necessario Rclone (vedi sopra), crea la cartella virtuale basata su Amazon S3 su Windows: (opzionale se tutti gli artefatti dell'applicazione provengono esclusivamente da Access). CodeCommit

### Note

Se hai già utilizzato Rclone durante l'anteprima di AWS Mainframe Modernization, devi eseguire l'aggiornamento alla versione più recente che si trova in. `m2-rclone.cmd` C:\Users\Public

1. Copia i `m2-rclone.cmd` file `m2-rclone.conf` and forniti nella tua cartella home usando File C:\Users\Public Explorer. C:\Users\PhotonUser\My Files\Home Folder
2. Aggiorna i parametri di `m2-rclone.conf` configurazione con la tua chiave di AWS accesso e il segreto corrispondente, oltre al tuo Regione AWS.

```
[m2-s3]
type = s3
provider = AWS
access_key_id = YOUR-ACCESS-KEY
secret_access_key = YOUR-SECRET-KEY
region = YOUR-REGION
acl = private
server_side_encryption = AES256
```

3. In `m2-rclone.cmd`, apporta le seguenti modifiche:
  - Cambia `your-s3-bucket` il nome del tuo bucket Amazon S3. Ad esempio, `m2-s3-mybucket`.
  - Passa `your-s3-folder-key` alla tua bucket key Amazon S3. Ad esempio, `myProject`.
  - Passa `your-local-folder-path` al percorso della directory in cui desideri sincronizzare i file dell'applicazione dal bucket Amazon S3 che li contiene. Ad esempio, `D:\PhotonUser\My Files\Home Folder\m2-new`. Questa directory sincronizzata deve essere una sottodirectory della cartella Home per consentire alla AppStream versione 2.0 di eseguire correttamente il backup e il ripristino all'inizio e alla fine della sessione.

```
:loop
timeout /T 10
"C:\Program Files\rclone\rclone.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rclone.conf"
goto :loop
```

4. Apri un prompt dei comandi di Windows, cd to C:\Users\PhotonUser\My Files\Home Folder se necessario ed esegui. m2-rclone.cmd Questo script di comandi esegue un ciclo continuo, sincronizzando il bucket e la chiave Amazon S3 con la cartella locale ogni 10 secondi. Puoi regolare il timeout in base alle tue esigenze. Dovresti vedere il codice sorgente dell'applicazione che si trova nel bucket Amazon S3 in Windows File Explorer.

Per aggiungere nuovi file al set su cui stai lavorando o aggiornare quelli esistenti, carica i file nel bucket Amazon S3 e verranno sincronizzati con la tua directory alla successiva iterazione definita in. m2-rclone.cmd Allo stesso modo, se desideri eliminare alcuni file, eliminali dal bucket Amazon S3. La prossima operazione di sincronizzazione li eliminerà dalla directory locale.

## Passaggio 3: clonare il repository

1. Vai al menu di selezione dell'applicazione nell'angolo in alto a sinistra della finestra del browser e seleziona Enterprise Developer.
2. Completa la creazione dell'area di lavoro richiesta da Enterprise Developer nella cartella Home scegliendo C:\Users\PhotonUser\My Files\Home Folder (aliasD: \PhotonUser\My Files\Home Folder) come posizione per l'area di lavoro.
3. In Enterprise Developer, clona il tuo CodeCommit repository andando su Project Explorer, fai clic con il pulsante destro del mouse e scegli Importa, Importa..., Git, Progetti dall'URI Git Clone. Quindi, inserisci le tue credenziali CodeCommit di accesso specifiche e completa la finestra di dialogo Eclipse per importare il codice.

Il repository CodeCommit git è ora clonato nel tuo spazio di lavoro locale.

L'area di lavoro di Enterprise Developer è ora pronta per iniziare i lavori di manutenzione dell'applicazione. In particolare, è possibile utilizzare l'istanza locale di Microfocus Enterprise Server (ES) integrata con Enterprise Developer per eseguire il debug in modo interattivo ed eseguire l'applicazione per convalidare le modifiche a livello locale.

### Note

L'ambiente Enterprise Developer locale, inclusa l'istanza locale di Enterprise Server, viene eseguito su Windows mentre AWS Mainframe Modernization viene eseguito su Linux. Si consiglia di eseguire test complementari nell'ambiente Linux fornito da AWS Mainframe Modernization dopo aver eseguito il commit della nuova applicazione CodeCommit e averla ricostruita per questo obiettivo e prima di implementarla in produzione.

## Sessioni successive

Quando selezioni una cartella gestita in modalità AppStream 2.0, ad esempio la cartella home per la clonazione del tuo CodeCommit repository, questa verrà salvata e ripristinata in modo trasparente tra le sessioni. La prossima volta che utilizzerai l'applicazione, completa i seguenti passaggi:

1. Avvia una sessione con AppStream 2.0 in base all'URL ricevuto nell'e-mail di benvenuto.
2. Accedi con la tua email e la password permanente.
3. Seleziona lo stack Enterprise Developer.
4. Avvia Rclone per connetterti (vedi sopra) al disco supportato da Amazon S3 quando questa opzione viene utilizzata per condividere i file dell'area di lavoro.
5. Avvia Enterprise Developer per fare il tuo lavoro.

## Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale in modo da non continuare a ricevere costi aggiuntivi. Completa questa procedura:

- Elimina il CodeCommit repository che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminare un CodeCommit repository](#) nella Guida per l'utente. AWS CodeCommit
- Elimina il database che hai creato per questo tutorial. Per ulteriori informazioni, consulta la sezione relativa all'[eliminazione di un'istanza database](#).



# Configurazione dell'automazione per le sessioni di streaming per sviluppatori Micro Focus Enterprise Analyzer e Micro Focus Enterprise Developer

È possibile eseguire automaticamente uno script all'inizio e alla fine della sessione per consentire un'automazione specifica per il contesto del cliente. Per ulteriori informazioni su questa funzionalità AppStream 2.0, consulta [Use Session Scripts to Manage Your AppStream 2.0 Users' Streaming Experience](#) nella Amazon AppStream 2.0 Administration Guide.

Questa funzionalità richiede che siano disponibili almeno le seguenti versioni delle immagini Enterprise Analyzer ed Enterprise Developer:

- `m2-enterprise-analyzer-v8.0.4.R1`
- `m2-enterprise-developer-v8.0.4.R1`

## Argomenti

- [Configura l'automazione all'avvio della sessione](#)
- [Configura l'automazione al termine della sessione](#)

## Configura l'automazione all'avvio della sessione

Se desideri eseguire uno script di automazione quando gli utenti si connettono alla AppStream versione 2.0, crea il tuo script e assegnagli un nome `m2-user-setup.cmd`. Memorizza lo script nella cartella Home AppStream 2.0 per l'utente. Le immagini AppStream 2.0 fornite da AWS Mainframe Modernization cercano uno script con quel nome in quella posizione e lo eseguono se esiste.

### Note

La durata dello script non può superare il limite impostato da AppStream 2.0, che attualmente è di 60 secondi. Per ulteriori informazioni, consulta [Eseguire gli script prima dell'inizio delle sessioni di streaming](#) nella Guida all'amministrazione di Amazon AppStream 2.0.

## Configura l'automazione al termine della sessione

Se desideri eseguire uno script di automazione quando gli utenti si disconnettono dalla AppStream versione 2.0, crea lo script e assegnagli un nome `mem2-user-teardown.cmd`. Memorizza lo script nella cartella Home AppStream 2.0 per l'utente. Le immagini AppStream 2.0 fornite da AWS Mainframe Modernization cercano uno script con quel nome in quella posizione e lo eseguono se esiste.

### Note

La durata dello script non può superare il limite impostato da AppStream 2.0, che attualmente è di 60 secondi. Per ulteriori informazioni, consulta [Esegui script dopo la fine delle sessioni di streaming](#) nella Guida all'amministrazione di Amazon AppStream 2.0.

## Visualizza i set di dati come tabelle e colonne in Enterprise Developer

È possibile accedere ai set di dati mainframe distribuiti in AWS Mainframe Modernization utilizzando il runtime Micro Focus. È possibile visualizzare i set di dati migrati come tabelle e colonne da un'istanza Micro Focus Enterprise Developer. La visualizzazione dei set di dati in questo modo consente di:

- Eseguire SQL `SELECT` operazioni sui file di dati migrati.
- Esponi i dati all'esterno dell'applicazione mainframe migrata senza modificare l'applicazione.
- Filtra facilmente i dati e salvali come CSV o altri formati di file.

### Note

I passaggi 1 e 2 sono attività una tantum. Ripeti i passaggi 3 e 4 per ogni set di dati per creare le viste del database.

### Argomenti

- [Prerequisiti](#)
- [Fase 1: Configurazione della connessione ODBC al datastore Micro Focus \(database Amazon RDS\)](#)

- [Fase 2: Creare il file MFDBFH.cfg](#)
- [Fase 3: Create un file di struttura \(STR\) per il layout del quaderno](#)
- [Fase 4: Creare una visualizzazione del database utilizzando il file di struttura \(STR\)](#)
- [Fase 5: Visualizzate i set di dati Micro Focus come tabelle e colonne](#)

## Prerequisiti

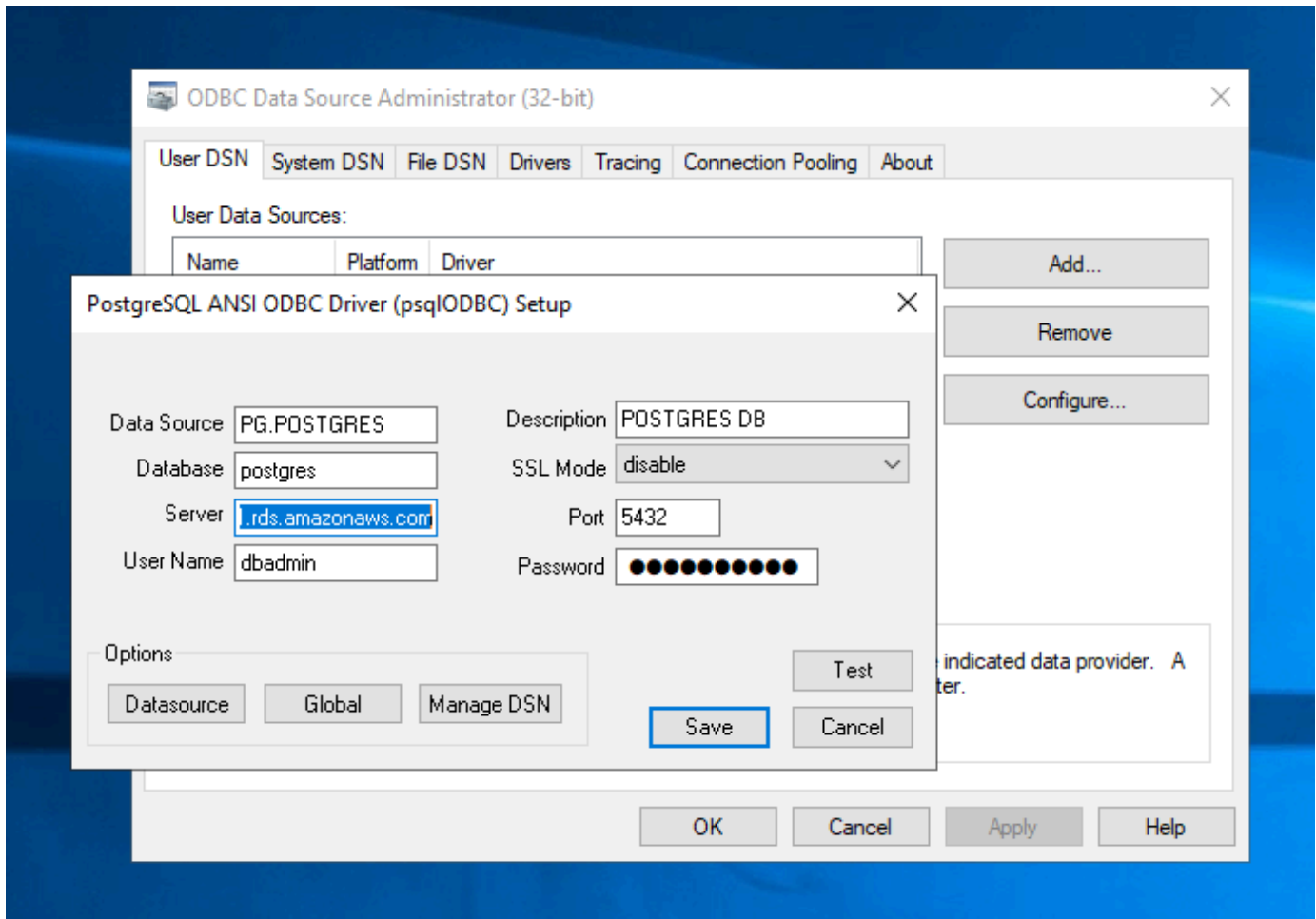
- È necessario avere accesso a Micro Focus Enterprise Developer Desktop tramite AppStream 2.0.
- È necessario che un'applicazione sia distribuita e funzionante in AWS Mainframe Modernization utilizzando il motore di runtime Micro Focus.
- Stai archiviando i dati dell'applicazione in Aurora PostgreSQL Compatible Edition.

## Fase 1: Configurazione della connessione ODBC al datastore Micro Focus (database Amazon RDS)

In questo passaggio, configuri una connessione ODBC al database che contiene i dati che desideri visualizzare come tabelle e colonne. Si tratta di un passaggio da eseguire una sola volta.

1. Accedete a Micro Focus Enterprise Developer Desktop utilizzando l'URL di streaming AppStream 2.0.
2. Apri ODBC Data Source Administrator, scegli User DSN, quindi scegli Aggiungi.
3. In Crea nuova origine dati, scegli PostgreSQL ANSI, quindi scegli Fine.
4. Crea una fonte di dati PG.POSTGRES fornendo le informazioni necessarie sul database, come segue:

```
Data Source : PG.POSTGRES
Database    : postgres
Server      : rds_endpoint.rds.amazonaws.com
Port        : 5432
User Name   : user_name
Password    : user_password
```



5. Scegli Test per assicurarti che la connessione funzioni. Dovresti vedere il messaggio Connection successful se il test ha esito positivo.

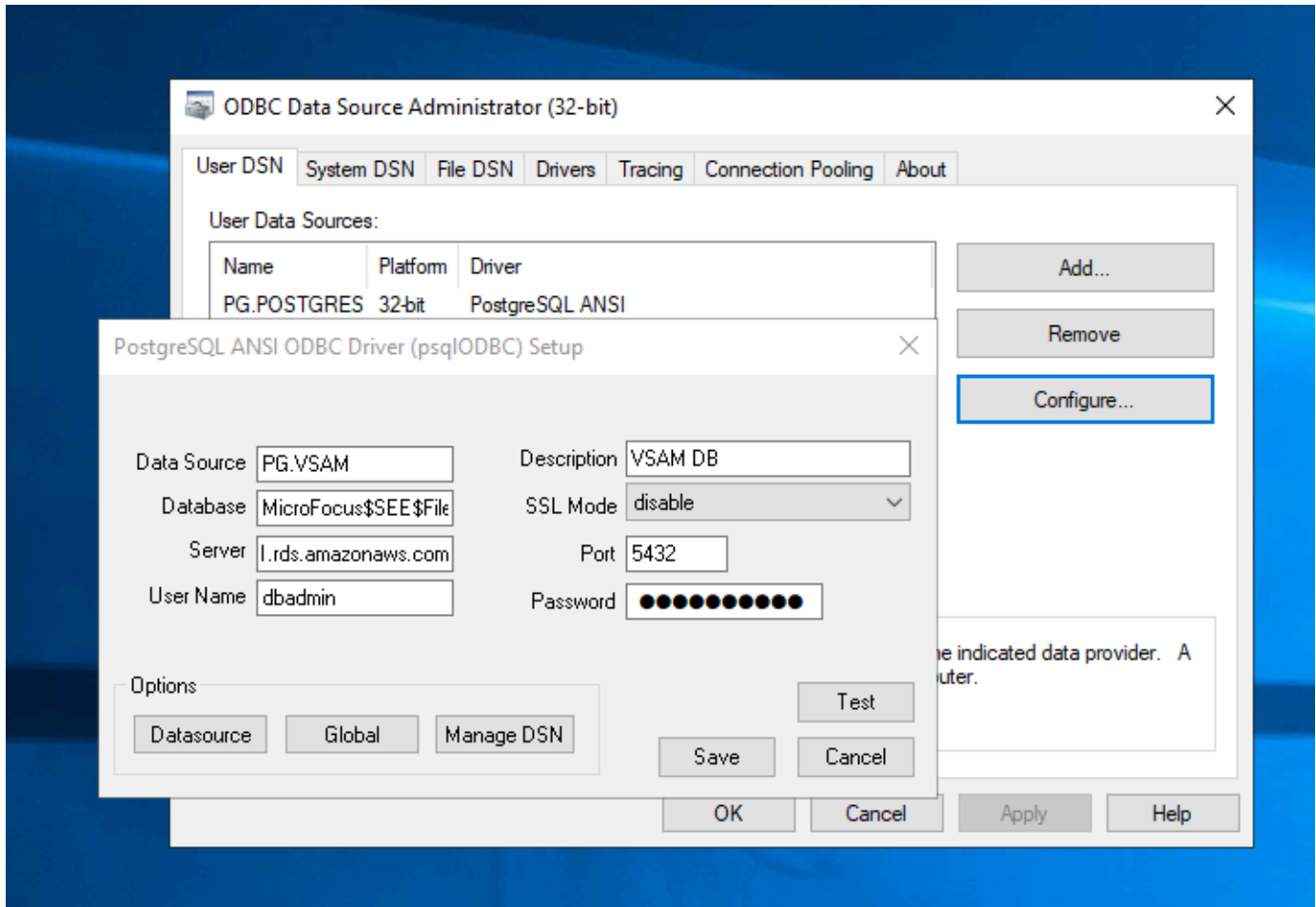
Se il test non ha esito positivo, consulta le seguenti informazioni.

- [Risoluzione dei problemi per Amazon RDS](#)
- [Come posso risolvere i problemi durante la connessione alla mia istanza database Amazon RDS?](#)

6. Salva la fonte di dati.
7. Crea un'origine dati perPG.VSAM, verifica la connessione e salva l'origine dati. Fornisci le seguenti informazioni sul database:

```
Data Source : PG.VSAM
Database    : MicroFocus$SEE$Files$VSAM
Server      : rds_endpoint.rds.amazonaws.com
Port        : 5432
```

User Name : *user\_name*  
 Password : *user\_password*



## Fase 2: Creare il file MFDBFH.cfg

In questo passaggio, si crea un file di configurazione che descrive il data store Micro Focus. Questa è una sola fase di configurazione.

1. Nella cartella Home, ad esempio, in `D:\PhotonUser\My Files\Home Folder\MFED\cfg\MFDBFH.cfg`, crea il file MFDBFH.cfg con il seguente contenuto.

```
<datastores>
  <server name="ESPACDatabase" type="postgresql" access="odbc">
    <dsn name="PG.POSTGRES" type="database" dbname="postgres"/>
    <dsn name="PG.VSAM" type="datastore" dsname="VSAM"/>
  </server>
```

```
</datastores>
```

2. Verificate la configurazione MFDBFH eseguendo i seguenti comandi per interrogare il datastore Micro Focus:

```
***  
*** Test the connection by running the following commands*  
***  
  
set MFDBFH_CONFIG="D:\PhotonUser\My Files\Home Folder\MFED\cfg\MFDBFH.cfg"  
  
dbfhdeploy list sql://ESPACDatabase/VSAM?folder=/DATA
```

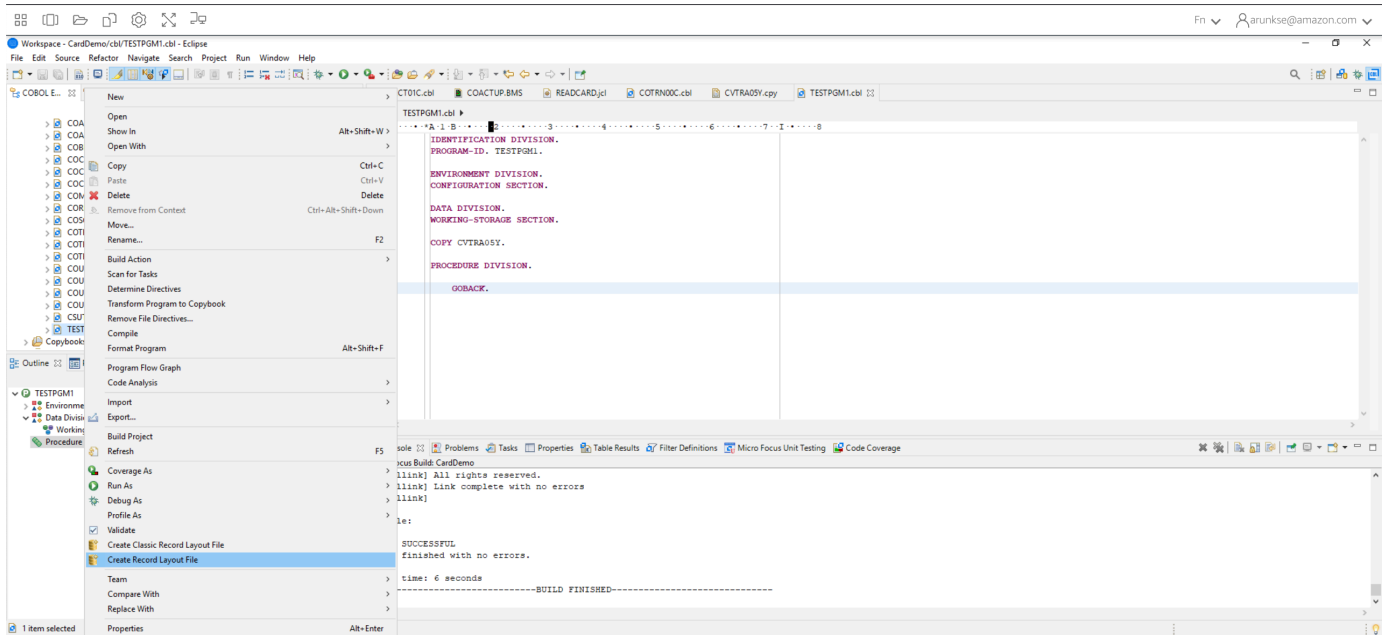
### Fase 3: Create un file di struttura (STR) per il layout del quaderno

In questo passaggio, creerai un file di struttura per il layout del tuo quaderno in modo da poterlo utilizzare in seguito per creare viste del database a partire dai set di dati.

1. Compila il programma associato al tuo quaderno. Se nessun programma utilizza il quaderno, crea e compila un programma semplice come il seguente con un'istruzione COPY per il tuo quaderno.

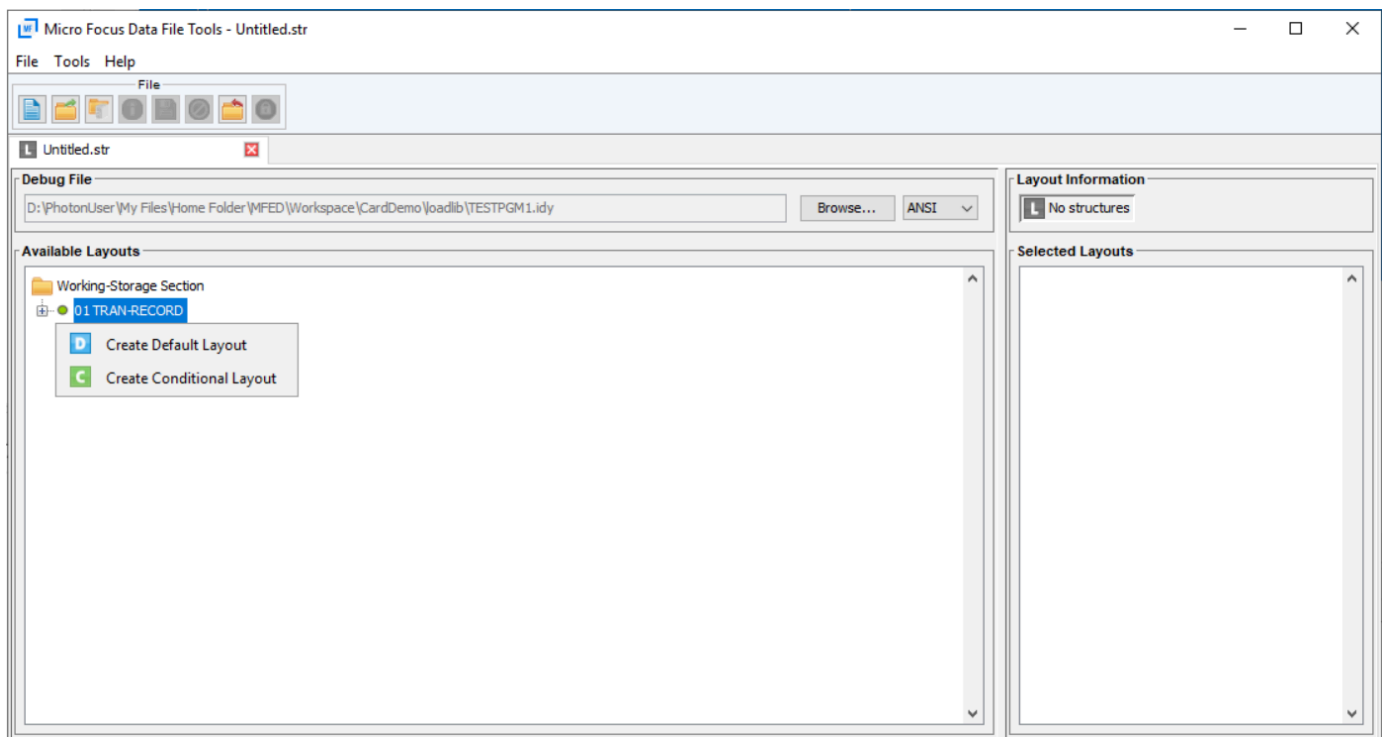
```
IDENTIFICATION DIVISION.  
PROGRAM-ID. TESTPGM1.  
  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
  
COPY CVTRA05Y.  
  
PROCEDURE DIVISION.  
  
GOBACK.
```

2. Dopo aver completato con successo la compilazione, fate clic con il pulsante destro del mouse sul programma e scegliete Create Record Layout File. Questo aprirà Micro Focus Data File Tools utilizzando il file.idy generato durante la compilazione.

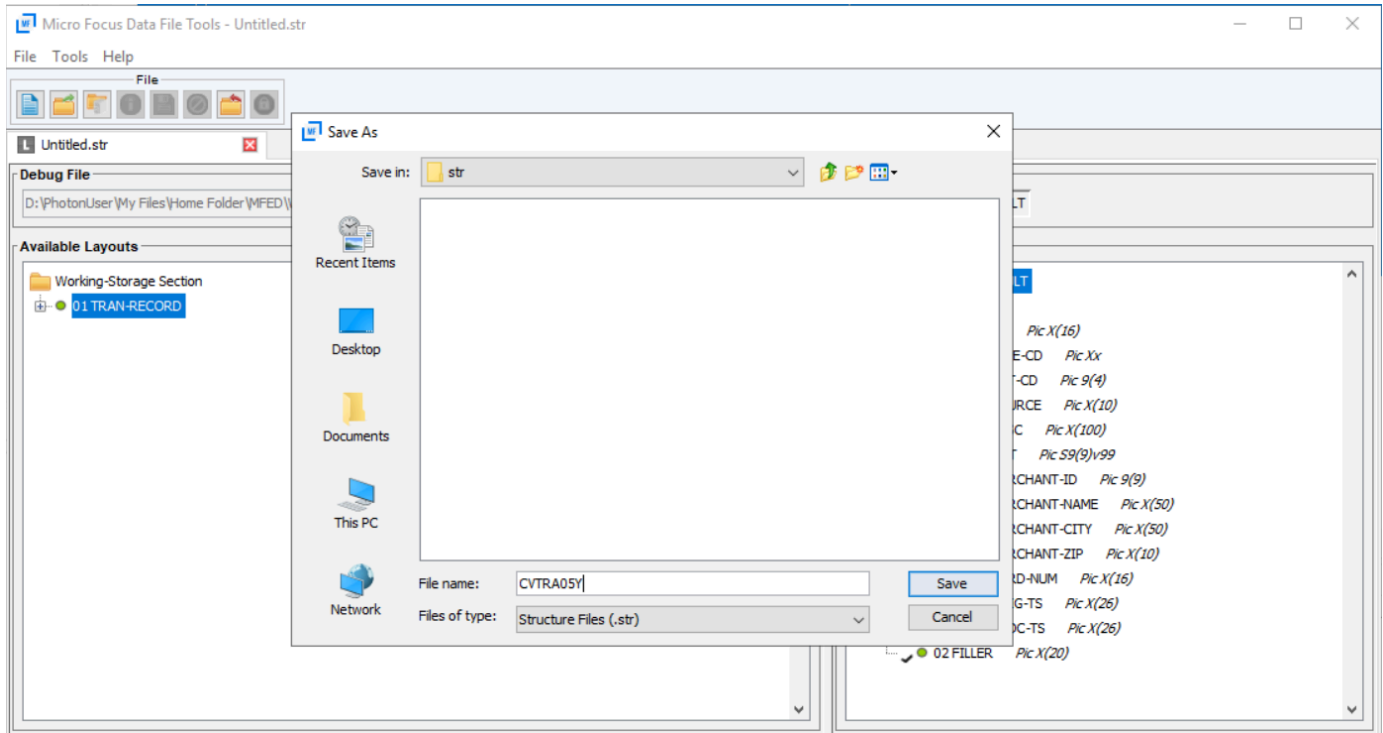


- Fate clic con il pulsante destro del mouse sulla struttura del record e scegliete Crea layout predefinito (struttura singola) o Crea layout condizionale (struttura multipla) a seconda del layout.

Per ulteriori informazioni, vedete [Creazione di file e layout di struttura](#) nella documentazione di Micro Focus.



4. Dopo aver creato il layout, scegliete File dal menu, quindi selezionate Salva con nome. Sfogliate e salva il file nella tua cartella Home con lo stesso nome del quaderno. Puoi scegliere di creare una cartella chiamata `str` e salvare lì tutti i tuoi file di struttura.



## Fase 4: Creare una visualizzazione del database utilizzando il file di struttura (STR)

In questo passaggio, si utilizza il file di struttura creato in precedenza per creare una vista del database per un set di dati.

- Utilizzate il `dbfhview` comando per creare una vista del database per un set di dati già presente nel datastore Micro Focus, come mostrato nell'esempio seguente.

```
##
## The below command creates database view for VSAM file
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS
## using the STR file CVTRA05Y.str
##

dbfhview -create -struct:"D:\PhotonUser\My Files\Home Folder\MFED\str
\CVTRA05Y.str" -name:V_AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT -file:sql://
ESPACDatabase/VSAM/AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT?folder=/DATA
```



```
##
## Output:
##
```

```
Micro Focus Database File Handler - View Generation Tool Version 8.0.00
Copyright (C) 1984-2022 Micro Focus. All rights reserved.
```

```
VGN0017I Using structure definition 'TRAN-RECORD-DEFAULT'
VGN0022I View 'V_AWS.M2.CARDDEMO.TRANSACTION.VSAM.KSDS.DAT' installed in
datastore 'sql://espacdatabase/VSAM'
VGN0002I The operation completed successfully
```

## Fase 5: Visualizzate i set di dati Micro Focus come tabelle e colonne

In questo passaggio, connettetevi al database utilizzando pgAdmin in modo da poter eseguire query per visualizzare i set di dati come tabelle e colonne.

- Connect al database MicroFocus\$SEE\$Files\$VSAM utilizzando pgAdmin e interroga la vista del database creata nel passaggio 4.

```
SELECT * FROM public."V_AWS.M2.CARDDEMO.TRANSACTION.VSAM.KSDS.DAT";
```

tran_id	tran_type_cd	tran_cat_cd	tran_source	tran_desc	tran_amt	tran_merchant_id	tran_merchant_name	tran_merchant_city	tran_merchant_zip	tran_card_num	tran_orig_ts	
1	000000000683580	01	0001	POS TERM	Purchase at Abshire-Lowe	0000005..	800000000	Abshire-Lowe	North Enoshaven	72112	485945261287...	2022-06-10
2	0000000001774250	03	0001	OPERATOR	Return item at Nitzsche, Nic...	0000009..	800000000	Nitzsche, Nicolas an...	Fidelshire	53378	092798710863...	2022-06-10
3	0000000006292564	01	0001	POS TERM	Purchase at Ermsler, Rob an...	0000000..	800000000	Ermsler, Rob and Gle...	North Malenkienzo...	78487-7965	609961915067...	2022-06-10
4	0000000009101861	01	0001	POS TERM	Purchase at Guann LLC	0000002..	800000000	Guann LLC	South Lynn	51508-9166	804058041034...	2022-06-10
5	00000000011042252	01	0001	POS TERM	Purchase at Kiertzmann-Scho...	0000004..	800000000	Kiertzmann-Schoen	East Eulastad	98754-1089	565683054498...	2022-06-10
6	00000000010229018	01	0001	POS TERM	Purchase at Gislason-Medhu...	0000008..	800000000	Gislason-Medhurst	Colleenburgh	23712-2080	737933563466...	2022-06-10
7	00000000016259484	03	0001	OPERATOR	Return item at Sipes Inc	0000000..	800000000	Sipes Inc	Emilioside	93329	401150089177...	2022-06-10
8	00000000017874199	01	0001	POS TERM	Purchase at Legros Group	0000003..	800000000	Legros Group	Cameloborough	34849-5127	804058041034...	2022-06-10
9	00000000019065428	03	0001	OPERATOR	Return item at Turcotte Group	0000005..	800000000	Turcotte Group	Andrewfurt	41346-3789	650353518179...	2022-06-10
10	00000000021711604	01	0001	POS TERM	Purchase at Gleason, Shana...	0000004..	800000000	Gleason, Shanahan a...	Myrticeport	21768-0823	950173732142...	2022-06-10
11	00000000025430891	01	0001	POS TERM	Purchase at Beatty-Hessel	0000000..	800000000	Beatty-Hessel	Simonisport	52595	326076361233...	2022-06-10
12	00000000028097268	01	0001	POS TERM	Purchase at Wolf, Cruicksha...	0000002..	800000000	Wolf, Cruickshank an...	Fritzchester	20195-5156	709414275105...	2022-06-10
13	00000000030752566	01	0001	POS TERM	Purchase at Ratke LLC	0000008..	800000000	Ratke LLC	Brendenfort	35302-4495	376628198415...	2022-06-10
14	00000000032979555	01	0001	POS TERM	Purchase at Treutel-Lefflar	0000000..	800000000	Treutel-Lefflar	New Nicolette	65014-0045	650923036255...	2022-06-10
15	00000000033688127	01	0001	POS TERM	Purchase at Schinner-Steuber	0000009..	800000000	Schinner-Steuber	Schmittchester	50777-5535	376628198415...	2022-06-10
16	00000000040455859	01	0001	POS TERM	Purchase at Brekke, Bradtke...	0000007..	800000000	Brekke, Bradtke and ...	Veummouth	18481-5013	114216769287...	2022-06-10
17	00000000043636099	03	0001	OPERATOR	Return item at Nader-Bayer	0000009..	800000000	Nader-Bayer	Goyetteville	35324	294013936230...	2022-06-10
18	00000000051205286	01	0001	POS TERM	Purchase at Goodwin, Von a...	0000006..	800000000	Goodwin, Von and Kr...	Erickmouth	03874	709414275105...	2022-06-10
19	00000000054298966	01	0001	POS TERM	Purchase at Cremin and Sons	0000005..	800000000	Cremin and Sons	Bartheleide	08677	453478410771...	7077-06-10

# Tutorial: Usare i modelli con Micro Focus Enterprise

Questo tutorial descrive come utilizzare modelli e progetti predefiniti con Micro Focus Enterprise Developer. Copre tre casi d'uso. Tutti i casi d'uso utilizzano il codice di esempio fornito nella BankDemo Esempio di Per scaricare il Esempio [bankdemo.zip](#).

## Important

Se si utilizza la versione di Enterprise Developer for Windows, i binari generati dal compilatore possono essere eseguiti solo su Enterprise Server fornito con Enterprise Developer. Non è possibile eseguirli sottoAWSRuntime di modernizzazione del mainframe, basato su Linux.

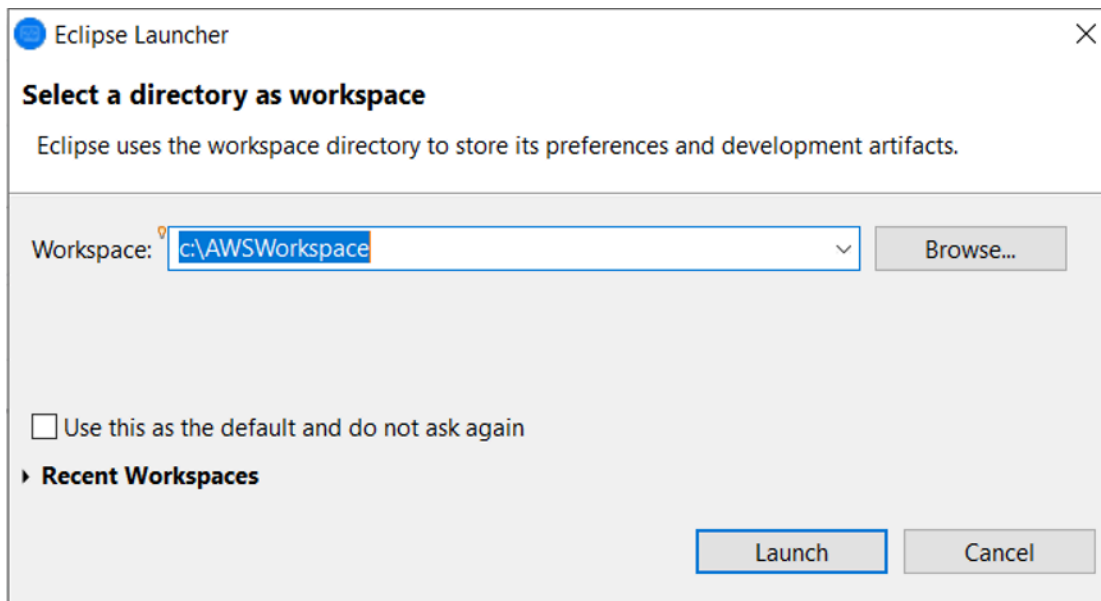
## Argomenti

- [Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine](#)
- [Caso d'uso 2: utilizzo del modello di progetto COBOL senza componenti di origine](#)
- [Caso d'uso 3 - Utilizzo del progetto COBOL predefinito per il collegamento alle cartelle di origine](#)
- [Utilizzo del modello JSON di definizione della regione](#)

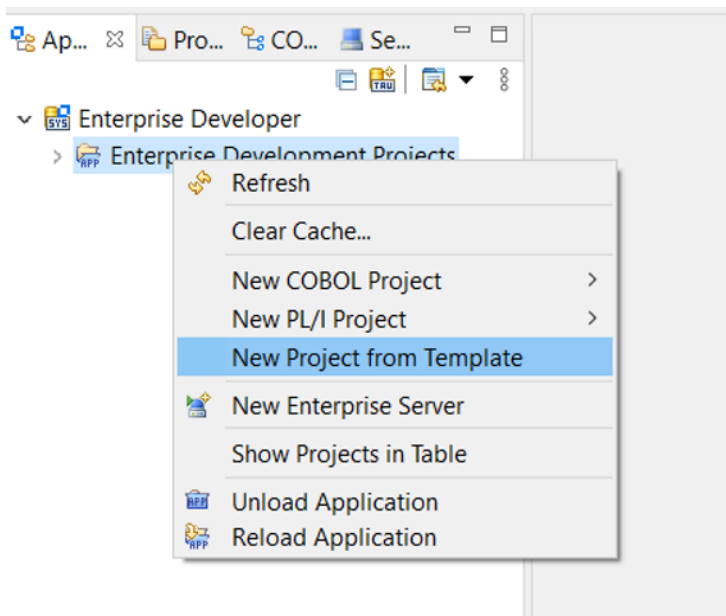
## Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine

Questo use case richiede la copia dei componenti di origine nella struttura di directory Template come parte delle fasi di pre-configurazione della demo. Nella [bankdemo.zip](#) questo è stato modificato rispetto all'originale `AWSTemplates.zip` consegna per evitare di avere due copie della fonte.

1. Avviare Enterprise Developer e specificare l'area di lavoro scelta.



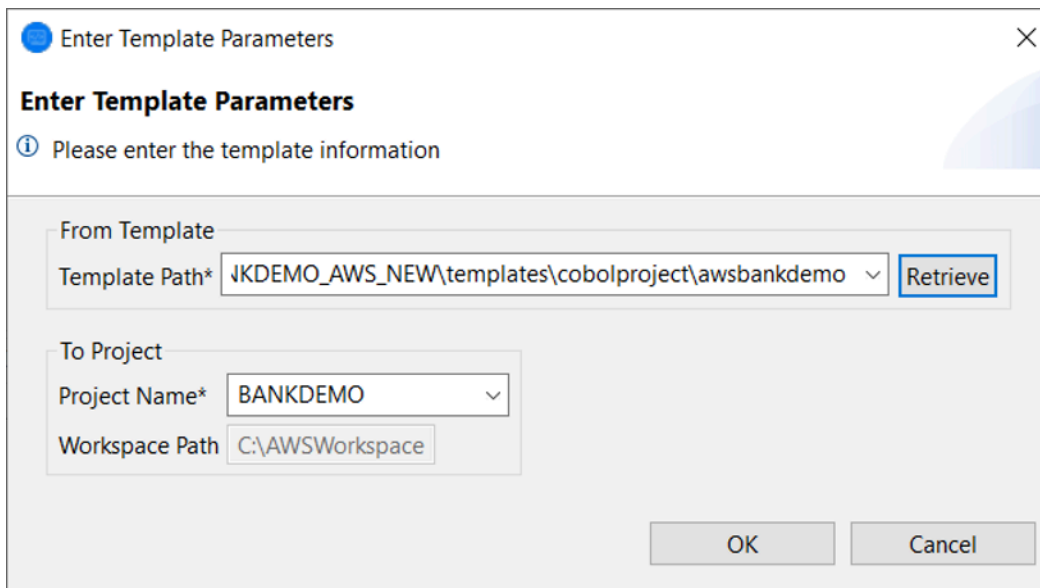
2. All'interno dell'Application Explorer vista, dal Progetto di sviluppo aziendale elemento della vista ad albero, scegliere Nuovo progetto da modello dal menu contestuale.



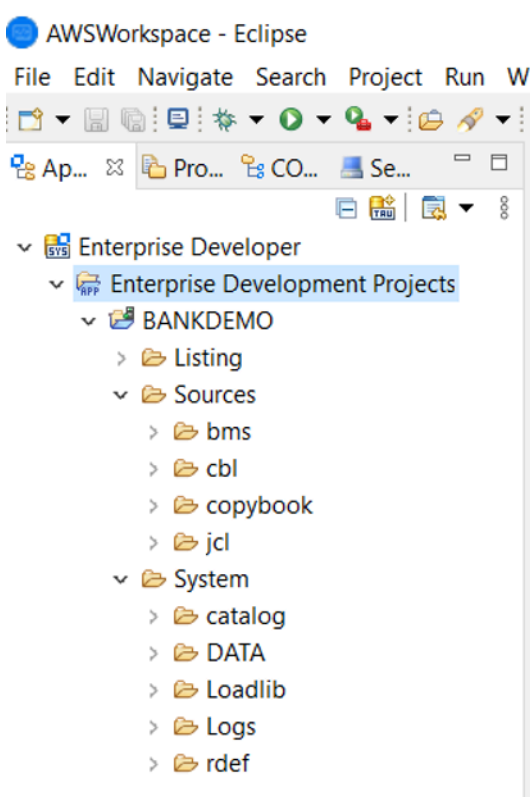
3. Immettere i parametri del modello come illustrato.

#### Note

Il percorso del modello si riferisce al punto in cui è stato estratto il file ZIP.



4. Scegliendo OK si creerà un progetto Eclipse di sviluppo locale basato sul modello fornito, con una struttura completa dell'ambiente di origine e di esecuzione.



LaSystemstruttura contiene un file di definizione delle risorse completo con le voci richieste per BANKDEMO, il catalogo richiesto con voci aggiunte e i corrispondenti file di dati ASCII.

Poiché la struttura del modello di origine contiene tutti gli elementi di origine, questi file vengono copiati nel progetto locale e pertanto vengono creati automaticamente in Enterprise Developer.

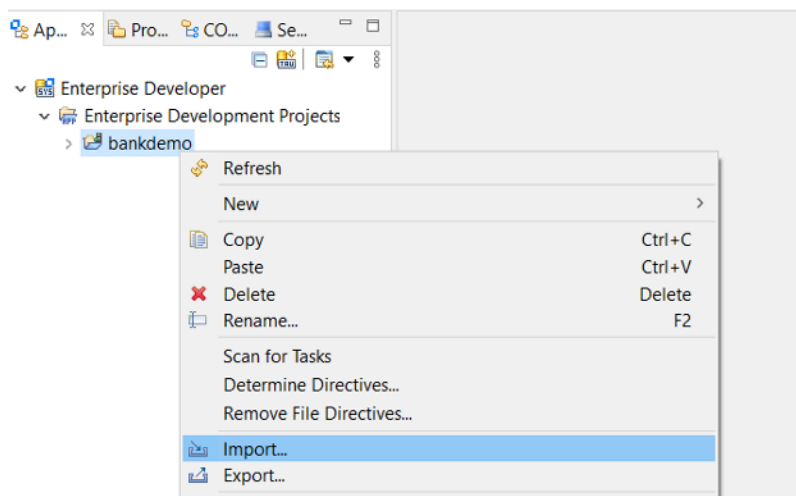
## Caso d'uso 2: utilizzo del modello di progetto COBOL senza componenti di origine

I passaggi da 1 a 3 sono identici a [Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine](#).

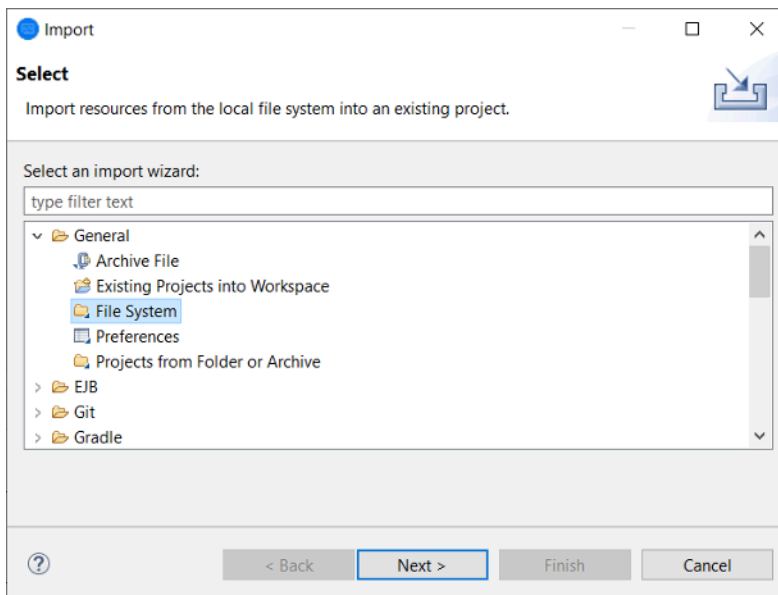
LaSystemstruttura in questo use case contiene anche un file di definizione delle risorse completo con le voci richieste per BankDemo, il catalogo richiesto con le voci aggiunte e i corrispondenti file di dati ASCII.

Tuttavia, la struttura di origine del modello non contiene alcun componente. È necessario importarli nel progetto da qualsiasi repository di origine in uso.

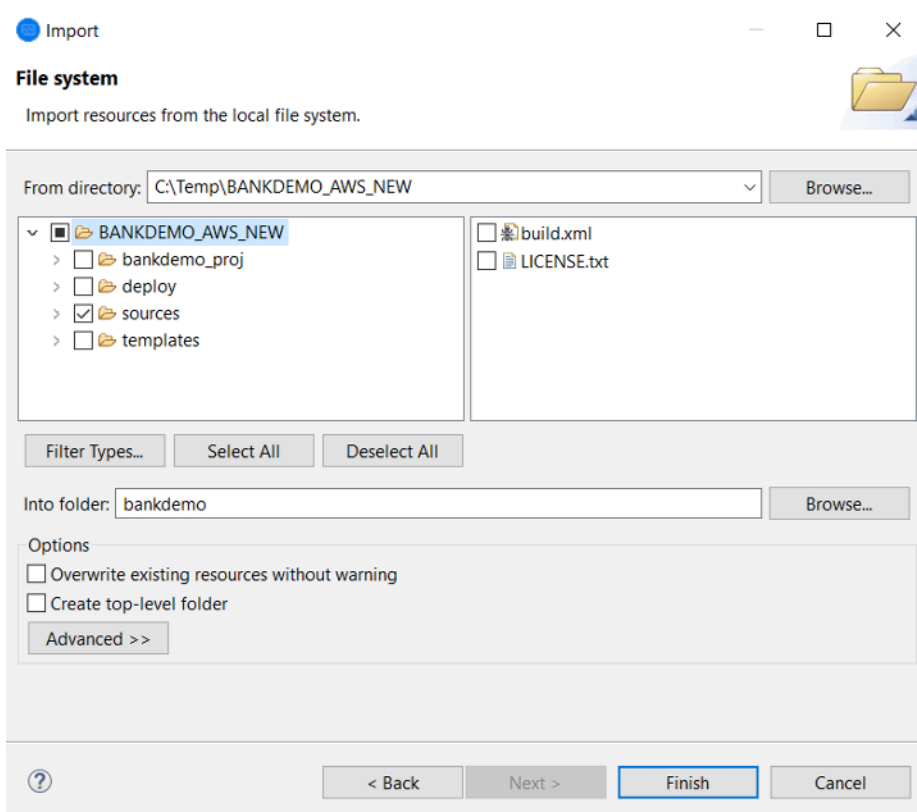
1. Scegli il nome del progetto. Nel relativo menu contestuale, selezionare **Importa**.



2. Dalla finestra di dialogo risultante, sotto **Generale** sezione, scegli **File system** (Successivo).



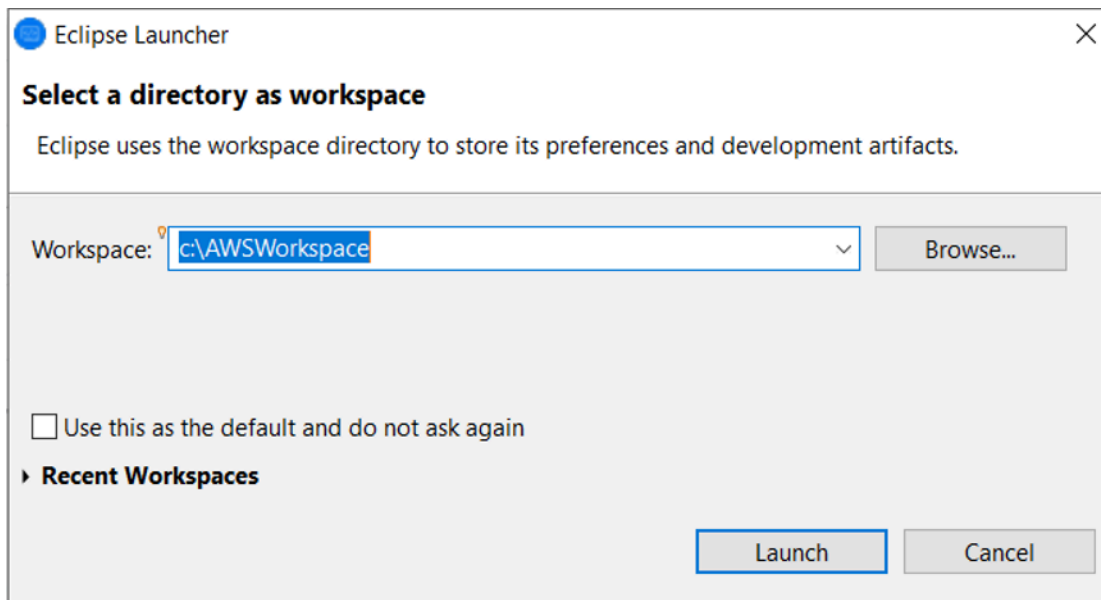
3. Popolamento del repository esplorando il file system in modo che punti alla cartella del repository. Scegli tutte le cartelle che desideri importare, ad esempio sources. La Into folder Il campo sarà precompilato. Scegli Finish (Fine).



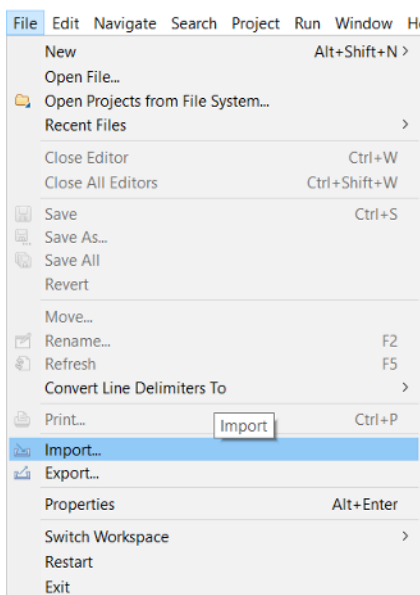
Dopo che la struttura del modello di origine contiene tutti gli elementi di origine, questi vengono creati automaticamente in Enterprise Developer.

## Caso d'uso 3 - Utilizzo del progetto COBOL predefinito per il collegamento alle cartelle di origine

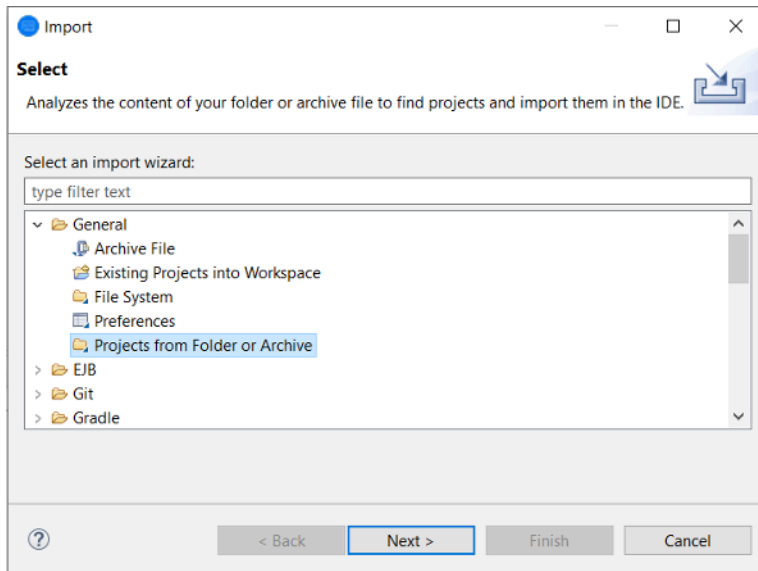
1. Avviare Enterprise Developer e specificare l'area di lavoro scelta.



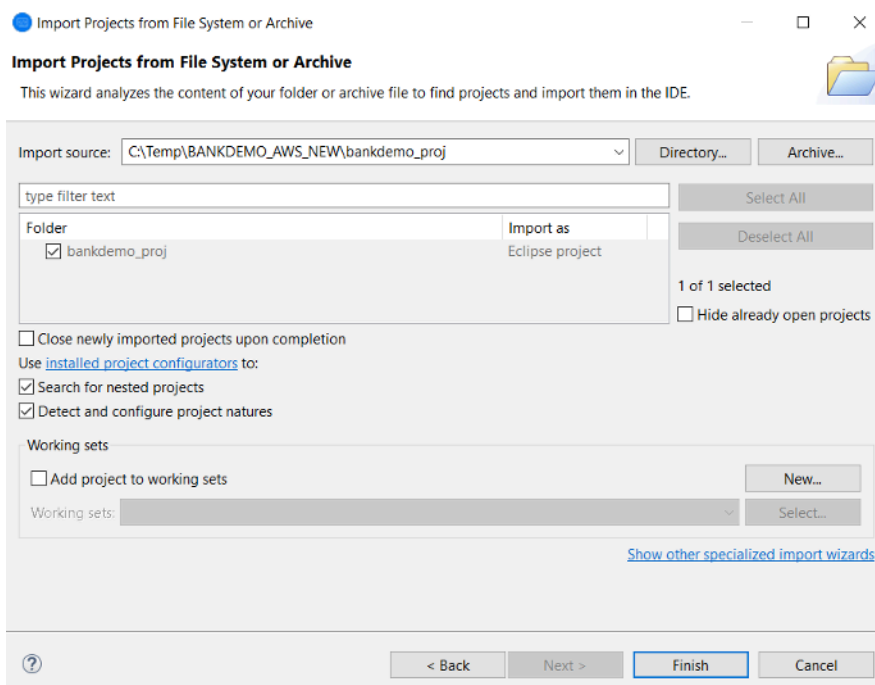
2. Dal menu File scegliere Import (Importa).



3. Dalla finestra di dialogo risultante, in Generale, scegli Progetti da cartella o archivio e scegli Successivo.



4. PopolamentoOrigine di importazione, ScegliereElencoe sfogliare il file system per selezionare la cartella di progetto predefinita. Il progetto contenuto in contiene collegamenti alle cartelle di origine nello stesso repository.



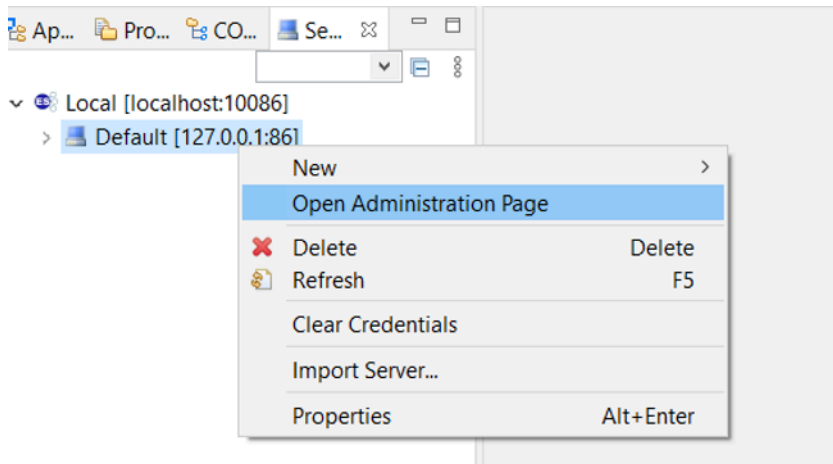
Scegli Finish (Fine).

Poiché il progetto è popolato dai collegamenti alla cartella di origine, il codice viene creato automaticamente.

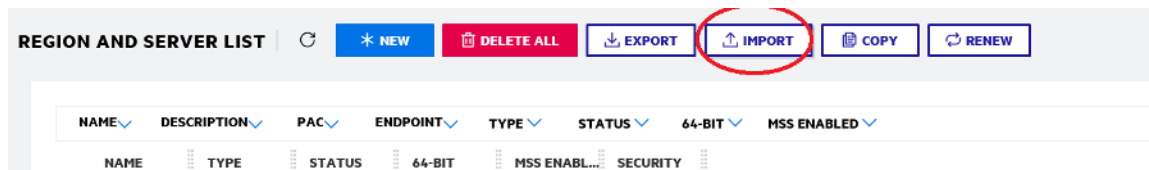


## Utilizzo del modello JSON di definizione della regione

1. Passa alla vista Server Explorer. Nel relativo menu contestuale, selezionare la pagina di amministrazione, che avvia il browser predefinito.



2. Dalla schermata risultante Enterprise Server Common Web Administration (ESCWA), scegliere Importa.



3. Seleziona JSON tipo di importazione e scelta Successivo.

### CHOOSE IMPORT TYPE



#### JSON

Import a .json file by selecting a file on the host where the client browser is running.

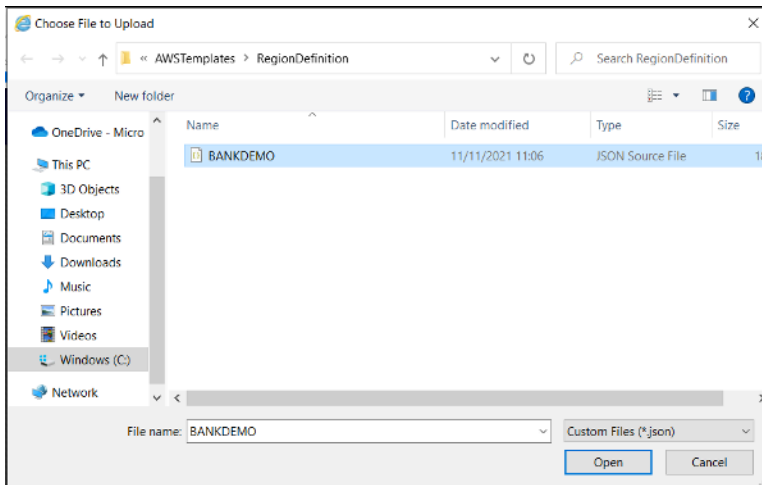
#### XML

Import a .xml file by selecting a file on the host where the client browser is running.

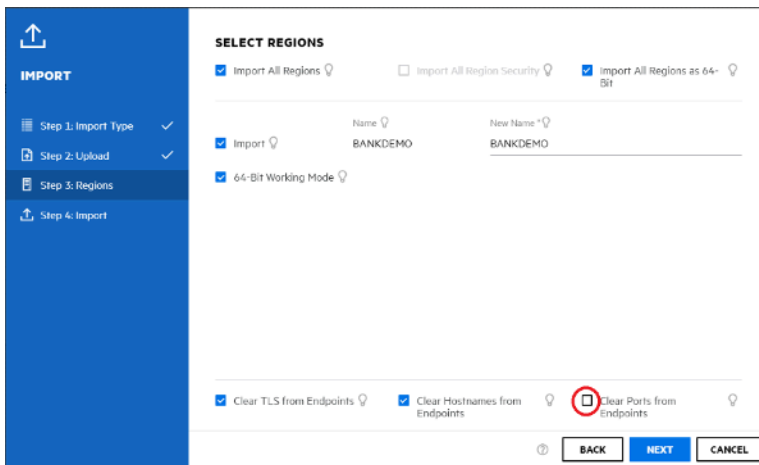
#### Legacy

Import a legacy repository (directory of .dat files) by selecting the directory location on the host where the Directory Server is running.

4. Carica il BANKDEMO.JSON file.



Una volta selezionato, scegli **Successivo**.



Sul **Seleziona regione** pannello, assicurarsi che **Cancella le porte dagli endpoint** L'opzione non è selezionata, quindi continua a scegliere **Successivo** attraverso i pannelli fino al **Esegui** importazione viene visualizzato il pannello. Scegliere quindi **Import (Importa)**.



Infine, fare clic **Termina**. La regione **BANKDEMO** verrà quindi aggiunta all'elenco dei server.

REGION AND SERVER LIST							
NAME	DESCRIPTION	PAC	ENDPOINT	TYPE	STATUS	64-BIT	MSS ENABLED
BANKDEMO	Region				Stopped		✓
ESDEMO	Region				Stopped		
ESDEMO64	Region				Stopped	✓	

5. Passare alla .Proprietà generaliper la regione BANKDEMO.
6. Scorri fino allaConfigurazioneSezione.
7. La variabile di ambiente ESP deve essere impostata suSystemcartella relativa al progetto Eclipse creata nei passaggi precedenti. Questo dovrebbe essereworkspacefolder/projectname/System.

```

ADDITIONAL

Configuration Information ⓘ
[ES-Environment]
ESP={Enter Project System Folder Here}
MF_CHARSET=A
EXTFH=$ESP/EXTFH.cfg

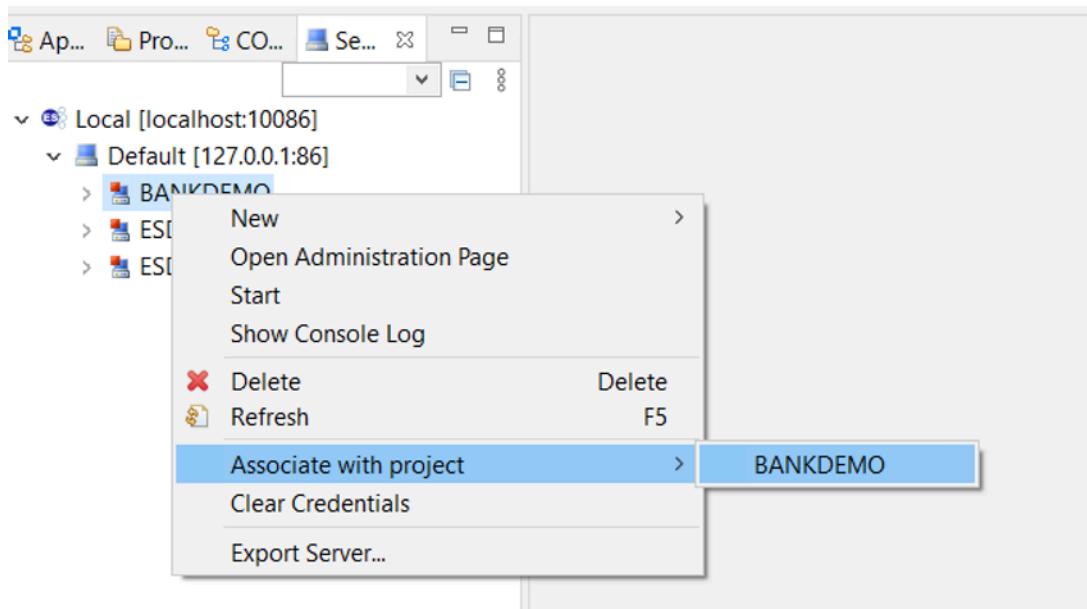
```

8. Fare clic su Apply (Applica).

APPLY

La regione è ora completamente configurata per essere eseguita insieme al progetto Eclipse COBOL.

9. Infine, tornando in Enterprise Developer, associa la regione importata al progetto.



L'ambiente Enterprise Developer è ora pronto per l'uso, con una versione funzionante completa di BankDemo. È possibile modificare, compilare ed eseguire il debug del codice rispetto alla regione.

#### Important

Se si utilizza la versione di Enterprise Developer for Windows, i binari generati dal compilatore possono essere eseguiti solo su Enterprise Server fornito con Enterprise Developer. Non è possibile eseguirli nel runtime di modernizzazione del mainframe, basato su Linux.

## Tutorial: configurazione della build Micro Focus per l'applicazione BankDemo di esempio

AWS La modernizzazione del mainframe offre la possibilità di configurare build e pipeline di integrazione continua/distribuzione continua (CI/CD) per le applicazioni migrate. Queste build e pipeline utilizzano e forniscono tali funzionalità. AWS CodeBuild AWS CodeCommit AWS CodePipeline CodeBuild è un servizio di compilazione completamente gestito che compila il codice sorgente, esegue test unitari e produce artefatti pronti per l'implementazione. CodeCommit è un servizio di controllo delle versioni che consente di archiviare e gestire in modo privato gli archivi Git

nel Cloud. AWS CodePipeline è un servizio di distribuzione continua che consente di modellare, visualizzare e automatizzare i passaggi necessari per il rilascio del software.

Questo tutorial dimostra come compilare il codice sorgente dell'applicazione di BankDemo esempio da Amazon S3 e quindi esportare il codice compilato in Amazon S3. AWS CodeBuild

AWS CodeBuild è un servizio di integrazione continua completamente gestito che permette di compilare codice sorgente, eseguire test e preparare pacchetti software pronti per essere distribuiti. Con CodeBuild, puoi utilizzare ambienti di compilazione preconfezionati oppure puoi creare ambienti di compilazione personalizzati che utilizzano i tuoi strumenti di compilazione. Questo scenario dimostrativo utilizza la seconda opzione. Consiste in un ambiente di CodeBuild compilazione che utilizza un'immagine Docker preconfezionata.

#### Important

Prima di iniziare il progetto di modernizzazione del mainframe, ti consigliamo di informarti sul [AWS Migration Acceleration Program \(MAP\) for Mainframe](#) o di [AWScontattare specialisti del mainframe](#) per conoscere i passaggi necessari per modernizzare un'applicazione mainframe.

## Argomenti

- [Prerequisiti](#)
- [Fase 1: creare bucket Amazon S3](#)
- [Passo 2: Crea il file delle specifiche di build](#)
- [Passaggio 3: carica i file sorgente](#)
- [Fase 4: Creare policy IAM](#)
- [Fase 5: Creare un ruolo IAM](#)
- [Fase 6: collegare le policy IAM al ruolo IAM](#)
- [Fase 7: Creare il progetto CodeBuild](#)
- [Fase 8: Avvia la compilazione](#)
- [Passaggio 9: scarica gli artefatti di output](#)
- [Pulizia delle risorse](#)

## Prerequisiti

Prima di iniziare questo tutorial, completate i seguenti prerequisiti.

- Scaricate l'[applicazione BankDemo di esempio](#) e decomprimetela in una cartella. La cartella di origine contiene i programmi COBOL e i Copybook e le definizioni CICS BMS. Contiene anche una cartella JCL come riferimento, sebbene non sia necessario creare JCL. La cartella contiene anche i meta file necessari per la compilazione.
- Nella console di modernizzazione del AWS mainframe, scegli Strumenti. In Analisi, sviluppo e creazione di risorse, scegli Condividi risorse con il mio account AWS.

## Fase 1: creare bucket Amazon S3

In questo passaggio, crei due bucket Amazon S3. Il primo è un bucket di input per contenere il codice sorgente e l'altro è un bucket di output per contenere l'output della build. Per ulteriori informazioni, consulta [Creazione, configurazione e utilizzo dei bucket Amazon S3 nella Amazon S3 User Guide](#).

1. Per creare il bucket di input, accedi alla console Amazon S3 e scegli Crea bucket.
2. Nella configurazione generale, fornisci un nome per il bucket e specifica Regione AWS dove desideri crearlo. Un esempio di nome è `codebuild-regionId-accountId-input-bucket`, `regionId` dov'è il Regione AWS bucket ed `accountId` è il tuo ID. Account AWS

### Note

Se state creando il bucket in un paese diverso Regione AWS dagli Stati Uniti orientali (Virginia settentrionale), specificate il parametro `LocationConstraint`. Per ulteriori informazioni, consulta [Create Bucket](#) nel riferimento all'API di Amazon Simple Storage Service.

3. Conserva tutte le altre impostazioni e scegli Crea bucket.
4. Ripeti i passaggi 1-3 per creare il bucket di output. Un esempio di nome è `codebuild-regionId-accountId-output-bucket`, dove `regionId` trova il Regione AWS bucket e `accountId` il tuo ID. Account AWS

Qualunque sia il nome che scegli per questi bucket, assicurati di usarli durante questo tutorial.

## Passo 2: Crea il file delle specifiche di build

In questo passaggio, crei un file di specifiche di build,. Questo file fornisce i comandi di compilazione e le relative impostazioni, in formato YAML, per CodeBuild eseguire la build. Per ulteriori informazioni, consulta il [riferimento alle specifiche della build CodeBuild nella Guida per l'AWS CodeBuildutente](#).

1. Crea un file denominato `buildspec.yml` nella directory che hai decompresso come prerequisito.
2. Aggiungi il seguente contenuto al file e salva. Non sono necessarie modifiche per questo file.

```
version: 0.2
env:
  exported-variables:
    - CODEBUILD_BUILD_ID
    - CODEBUILD_BUILD_ARN
phases:
  install:
    runtime-versions:
      python: 3.7
  pre_build:
    commands:
      - echo Installing source dependencies...
      - ls -lR $CODEBUILD_SRC_DIR/source
  build:
    commands:
      - echo Build started on `date`
      - /start-build.sh -Dbasedir=$CODEBUILD_SRC_DIR/source -Dloaddir=
$CODEBUILD_SRC_DIR/target
  post_build:
    commands:
      - ls -lR $CODEBUILD_SRC_DIR/target
      - echo Build completed on `date`
artifacts:
  files:
    - $CODEBUILD_SRC_DIR/target/**
```

Qui `CODEBUILD_BUILD_ID`, `CODEBUILD_BUILD_ARN`, `$CODEBUILD_SRC_DIR/source`, e `$CODEBUILD_SRC_DIR/target` ci sono variabili di ambiente disponibili all'interno CodeBuild. Per ulteriori informazioni, consulta [Variabili di ambiente negli ambienti di compilazione](#).

A questo punto, la tua directory dovrebbe avere questo aspetto.

```
(root directory name)
|-- build.xml
|-- buildspec.yml
|-- LICENSE.txt
|-- source
```

```
|... etc.
```

3. Comprimi il contenuto della cartella in un file chiamato `BankDemo.zip`. Per questo tutorial, non puoi comprimere la cartella. Invece, comprimi il contenuto della cartella nel file `BankDemo.zip`.

## Passaggio 3: carica i file sorgente

In questo passaggio, carichi il codice sorgente per l'applicazione di `BankDemo` esempio nel tuo bucket di input Amazon S3.

1. Accedi alla console Amazon S3 e scegli `Bucket` nel riquadro di navigazione a sinistra. Quindi scegli il bucket di input che hai creato in precedenza.
2. In `Oggetti`, scegli `Carica`.
3. Nella sezione `File e cartelle`, scegli `Aggiungi file`.
4. Vai al tuo `BankDemo.zip` file e scegli.
5. Scegli `Carica`.

## Fase 4: Creare policy IAM

In questo passaggio, crei due [policy IAM](#). Una policy concede le autorizzazioni per `AWS Mainframe Modernization` per accedere e utilizzare l'immagine `Docker` che contiene gli strumenti di compilazione di `Micro Focus`. Questa politica non è personalizzata per i clienti. [L'altra politica concede le autorizzazioni per AWS Mainframe Modernization per interagire con i bucket di input e output e con i log Amazon generati. CloudWatch CodeBuild](#)

Per ulteriori informazioni sulla creazione di una policy IAM, consulta [Modifica delle politiche IAM nella IAM User Guide](#).

Per creare una policy per l'accesso alle immagini `Docker`

1. Nella console IAM, copia il seguente documento di policy e incollalo nell'editor delle policy.

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



```

        "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": [
      "ecr:BatchCheckLayerAvailability",
      "ecr:GetDownloadUrlForLayer",
      "ecr:BatchGetImage"
    ],
    "Resource": "arn:aws:ecr:*:673918848628:repository/m2-enterprise-build-
tools"
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::aws-m2-repo-*/*"
  }
]
}

```

2. Fornisci un nome per la policy, ad esempio `m2CodeBuildPolicy`.

Creare una policy che consenta alla modernizzazione AWS del mainframe di interagire con bucket e log

1. Nella console IAM, copia il seguente documento di policy e incollalo nell'editor delle policy. Assicurati di eseguire l'aggiornamento `regionId` a Regione AWS, e `accountId` al tuo Account AWS.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
    },
  ],
}

```

```

        "Resource": [
            "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/
codebuild-bankdemo-project",
            "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/
codebuild-bankdemo-project:*"
        ],
        "Effect": "Allow"
    },
    {
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:GetObjectVersion",
            "s3:GetBucketAcl",
            "s3:GetBucketLocation",
            "s3:List*"
        ],
        "Resource": [
            "arn:aws:s3:::codebuild-regionId-accountId-input-bucket",
            "arn:aws:s3:::codebuild-regionId-accountId-input-bucket/*",
            "arn:aws:s3:::codebuild-regionId-accountId-output-bucket",
            "arn:aws:s3:::codebuild-regionId-accountId-output-bucket/*"
        ],
        "Effect": "Allow"
    }
]
}

```

2. Fornisci un nome per la politica, ad esempio, BankdemoCodeBuildRolePolicy.

## Fase 5: Creare un ruolo IAM

In questo passaggio, crei un nuovo [ruolo IAM](#) che CodeBuild consenta di interagire con AWS le risorse per te, dopo aver associato le policy IAM che hai creato in precedenza a questo nuovo ruolo IAM.

Per informazioni sulla creazione di un ruolo di servizio, consulta [Creating a Role to Delegate Permissions to an AWS Service](#) nella IAM User Guide,.

1. Accedi alla console IAM e scegli Ruoli nel riquadro di navigazione a sinistra.
2. Scegli Crea ruolo.

3. In Tipo di entità affidabile, scegli il servizio AWS.
4. In Casi d'uso per altri servizi AWS CodeBuild, scegli, quindi scegli di CodeBuildnuovo.
5. Scegli Next (Successivo).
6. Nella pagina Add permissions (Aggiungi autorizzazioni), scegli Next (Successivo). Successivamente assegnerai una policy al ruolo.
7. In Dettagli del ruolo, fornisci un nome per il ruolo, BankdemoCodeBuildServiceRole ad esempio.
8. In Seleziona entità attendibili, verifica che il documento di policy abbia il seguente aspetto:

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

9. Scegli Crea ruolo.

## Fase 6: collegare le policy IAM al ruolo IAM

In questo passaggio, colleghi le due policy IAM create in precedenza al ruolo BankdemoCodeBuildServiceRole IAM.

1. Accedi alla console IAM e scegli Roles nel riquadro di navigazione a sinistra.
2. In Ruoli, scegli il ruolo che hai creato in precedenza, ad esempio BankdemoCodeBuildServiceRole.
3. In Criteri di autorizzazione, scegli Aggiungi autorizzazioni, quindi Allega criteri.
4. In Altre politiche di autorizzazione, scegli le politiche che hai creato in precedenza, ad esempio e. m2CodeBuildPolicy BankdemoCodeBuildRolePolicy
5. Scegli Collega policy.

## Fase 7: Creare il progetto CodeBuild

In questo passaggio, crei il CodeBuild progetto.

1. Accedi alla CodeBuild console e scegli Crea progetto di compilazione.
2. Nella sezione Configurazione del progetto, fornisci un nome per il progetto, ad esempio `codebuild-bankdemo-project`.
3. Nella sezione Sorgente, per Provider di origine, scegli Amazon S3, quindi scegli il bucket di input creato in precedenza, ad esempio `codebuild-regionId-accountId-input-bucket`
4. Nel campo della chiave dell'oggetto S3 o della cartella S3, inserisci il nome del file zip che hai caricato nel bucket S3. In questo caso, il nome del file è `bankdemo.zip`
5. Nella sezione Ambiente, scegli Immagine personalizzata.
6. Nel campo Tipo di ambiente, scegli Linux.
7. In Registro delle immagini, scegli Altro registro.
8. Nel campo URL del registro esterno, inserisci `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:latest`
9. In Ruolo di servizio, scegli Ruolo di servizio esistente e, nel campo Role ARN, scegli il ruolo di servizio creato in precedenza, ad esempio `BankdemoCodeBuildServiceRole`
10. Nella sezione Buildspec, scegli Usa un file buildspec.
11. Nella sezione Artefatti, in Tipo, scegli Amazon S3, quindi scegli il tuo bucket di output, ad esempio `codebuild-regionId-accountId-output-bucket`
12. Nel campo Nome, inserisci il nome di una cartella nel bucket in cui desideri contenere gli artefatti di output della build, ad esempio `bankdemo-output.zip`
13. In Artifacts packaging, scegli Zip.
14. Scegliere Create build project (Crea progetto di compilazione).

## Fase 8: Avvia la compilazione

In questo passaggio, si avvia la compilazione.

1. Accedi alla CodeBuild console.
2. Nel riquadro di navigazione a sinistra, scegli Crea progetti.
3. Scegli il progetto di compilazione che hai creato in precedenza, ad esempio `codebuild-bankdemo-project`.

#### 4. Selezionare Start build (Avvia compilazione).

Questo comando avvia la compilazione. La build viene eseguita in modo asincrono. L'output del comando è un JSON che include l'attributo id. Questo attributo id è un riferimento all'id di CodeBuild build della build appena avviata. Puoi visualizzare lo stato della build nella CodeBuild console. Puoi anche visualizzare i log dettagliati sull'esecuzione della build nella console. Per ulteriori informazioni, consulta [Visualizza informazioni dettagliate sulla build](#) nella Guida per l'AWS CodeBuild utente.

Quando la fase corrente è COMPLETATA, significa che la build è stata completata correttamente e gli artefatti compilati sono pronti su Amazon S3.

### Passaggio 9: scarica gli artefatti di output

In questo passaggio, scarichi gli artefatti di output da Amazon S3. Lo strumento di compilazione Micro Focus può creare diversi tipi di eseguibili. In questo tutorial, genera oggetti condivisi.

1. Accedi alla console Amazon S3.
2. Nella sezione Bucket, scegli il nome del tuo bucket di output, ad esempio. `codebuild-regionId-accountId-output-bucket`
3. Scegli Download (Scarica).
4. Decomprimere il file scaricato. Vai alla cartella di destinazione per vedere gli artefatti della build. Questi includono gli oggetti condivisi di `.so` Linux.

### Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale per evitare costi aggiuntivi. Per fare ciò, completa la seguente procedura:

- Elimina i bucket S3 che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
- Elimina le policy IAM che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminazione delle policy IAM](#) nella IAM User Guide.
- Elimina il ruolo IAM che hai creato per questo tutorial. Per ulteriori informazioni, consulta la sezione [Eliminazione di ruoli o profili delle istanze](#) nella Guida per l'utente di IAM.
- Elimina il CodeBuild progetto che hai creato per questo tutorial. Per ulteriori informazioni, consulta [Eliminare un progetto di compilazione CodeBuild nella](#) Guida AWS CodeBuild per l'utente.

# Tutorial: configurazione di una pipeline CI/CD da utilizzare con Micro Focus Enterprise Developer

Questo tutorial mostra come importare, modificare, compilare ed eseguire l'applicazione di BankDemo esempio in Micro Focus Enterprise Developer e quindi come eseguire il commit delle modifiche per attivare una pipeline CI/CD.

## Indice

- [Prerequisiti](#)
- [Crea l'infrastruttura di base della pipeline CI/CD](#)
- [Crea un repository e una pipeline CI/CD AWS CodeCommit](#)
  - [Esempio di file Trigger YAML config\\_git.yml](#)
- [Creazione di AppStream Enterprise Developer 2.0](#)
- [Configurazione e test di Enterprise Developer](#)
  - [Clonare il BankDemo CodeCommit repository in Enterprise Developer](#)
  - [Crea un progetto COBOL BankDemo mainframe e crea un'applicazione](#)
  - [Crea un ambiente BankDemo CICS e batch locale per i test](#)
  - [Avvia il server BANKDEMO da Enterprise Developer](#)
  - [Avvia il terminale Rumba 3270](#)
  - [Esegui una BankDemo transazione](#)
  - [Arresta il server BANKDEMO da Enterprise Developer](#)
- [Esercizio 1: Migliora il calcolo del prestito nell'applicazione BANKDEMO](#)
  - [Aggiungi la regola di analisi del prestito a Enterprise Developer Code Analysis](#)
  - [Fase 1: Eseguire l'analisi del codice per il calcolo del prestito](#)
  - [Passaggio 2: modifica della mappa CICS BMS e del programma COBOL e test](#)
  - [Fase 3: Aggiungere il calcolo dell'importo totale nel programma COBOL](#)
  - [Fase 4: Confermare le modifiche ed eseguire la pipeline CI/CD](#)
- [Esercizio 2: Estrarre il calcolo del prestito nella domanda BankDemo](#)
  - [Fase 1: rifattorizzare la routine di calcolo dei prestiti in una sezione COBOL](#)
  - [Fase 2: Estrarre la routine di calcolo del prestito in un programma COBOL autonomo](#)
  - [Fase 3: Conferma delle modifiche ed esecuzione della pipeline CI/CD](#)

- [Pulizia delle risorse](#)

## Prerequisiti

Scaricate i seguenti file.

- `basic-infra.yaml`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).
- `pipeline.yaml`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).
- `m2-code-sync-function.zip`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).
- `config_git.yaml`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).
- `BANKDEMO-source.zip`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).
- `BANKDEMO-exercise.zip`
  - [Scarica dalla regione Europa \(Francoforte\)](#).
  - [Scarica dalla regione Stati Uniti orientali \(Virginia settentrionale\)](#).

Lo scopo di ogni file è il seguente:

`basic-infra.yaml`

Questo AWS CloudFormation modello crea l'infrastruttura di base necessaria per la pipeline CI/CD: VPC, bucket Amazon S3 e così via.

## pipeline.yaml

Questo AWS CloudFormation modello viene utilizzato da una funzione Lambda per avviare lo stack di pipeline. Assicurati che questo modello si trovi in un bucket Amazon S3 accessibile al pubblico. Aggiungi il link a questo bucket come valore predefinito per il PipelineTemplateURL parametro nel modello. `basic-infra.yaml`

## m2-code-sync-function.zip

Questa funzione Lambda crea il CodeCommit repository, la struttura di directory basata su `config_git.yaml`, e avvia lo stack di pipeline utilizzando `pipeline.yaml`. Assicurati che questo file zip sia disponibile in un bucket Amazon S3 accessibile al pubblico in tutti i paesi in Regioni AWS cui è supportata la modernizzazione del AWS mainframe. Ti consigliamo di archiviare il file in un unico bucket Regione AWS e di replicarlo in tutti i bucket. Regioni AWS Utilizzate una convenzione di denominazione per il bucket con un suffisso che identifichi lo specifico Regione AWS (ad esempio `m2-cicd-deployment-source-eu-west-1`) e aggiungete il prefisso `m2-cicd-deployment-source` come valore predefinito per il parametro `DeploymentSourceBucket` e formate il bucket completo utilizzando la funzione `!Sub {DeploymentSourceBucket}-${AWS::Region}` di AWS CloudFormation sostituzione mentre fate riferimento a quel bucket nel modello di risorsa. `basic-infra.yaml`  
`SourceSyncLambdaFunction`

## config\_git.yml

CodeCommit definizione della struttura delle cartelle. Per ulteriori informazioni, consulta [Esempio di file Trigger YAML config\\_git.yml](#).

## BANKDEMO-source.zip.

BankDemo codice sorgente e file di configurazione creati dal CodeCommit repository.

## BANKDEMO-exercise.zip.

BankDemo fonte per gli esercizi tutorial creati dal CodeCommit repository.

## Crea l'infrastruttura di base della pipeline CI/CD

Usa il AWS CloudFormation modello `basic-infra.yaml` per creare lo stack di infrastruttura di base della pipeline CI/CD tramite la console. AWS CloudFormation Questo stack crea bucket Amazon S3 in cui caricare il codice e i dati dell'applicazione e una funzione di AWS Lambda supporto per creare altre risorse necessarie come un repository e AWS CodeCommit una pipeline. AWS CodePipeline



**Note**

Per avviare questo stack sono necessarie le autorizzazioni per amministrare IAM, Amazon S3, Lambda e le autorizzazioni per l'uso. AWS CloudFormation AWS KMS

1. Accedere alla AWS Management Console e aprire la console di AWS CloudFormation all'indirizzo <https://console.aws.amazon.com/cloudformation>.
2. Creare un nuovo stack utilizzando una delle seguenti opzioni:
  - Scegli Crea stack. Questa è la sola opzione se si dispone di uno stack attualmente in esecuzione.
  - Nella pagina Stacks, scegli Crea stack. Questa opzione è visibile solo se non esistono stack in esecuzione.
3. Nella pagina Specificare il modello:
  - In Prepara il modello, scegli Il modello è pronto.
  - In Specificare modello, scegli l'URL di Amazon S3 come origine del modello e inserisci uno dei seguenti URL a seconda del tuo. Regione AWS
    - `https://m2-us-east-1.s3.us-east-1.amazonaws.com/cicd/mf/basic-infra.yaml`
    - `https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/basic-infra.yaml`
  - Per accettare le tue impostazioni, scegli Avanti.

Viene visualizzata la pagina Crea stack.

## Specify stack details

### Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

### Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

#### Networking Configuration

Do you want to use an existing VPC in your account?

If you select 'Yes', then you must provide the VPC ID and the Subnet IDs.

Which VPC ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID in a different AZ should be used for HA?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Enter the CIDR block that should be used for the new VPC

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

CIDR bits for creating subnets. Choose 5 for /27, 6 for /26, 7 for /25, 8 for /24 range

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

#### Deployment Configuration

Name of the S3 bucket which contains the source files for this stack deployment

Don't change unless you know what you are doing.

Name of the source package file for the infrastructure Lambda function

Don't change unless you know what you are doing.

Full URL of the pipeline CloudFormation template file


Don't change unless you know what you are doing.

What name prefix to use for the new S3 buckets?

A name prefix for the S3 buckets that will be created by this stack.


Apporta le seguenti modifiche:

- Fornite i valori appropriati per il nome dello stack e i parametri per la configurazione di rete.
- La maggior parte dei parametri nelle configurazioni di distribuzione sono precompilati in modo appropriato, quindi non è necessario modificarli. A seconda del modello Regione AWS, modifica il AWS CloudFormation modello di pipeline con uno dei seguenti URL Amazon S3.
  - <https://m2-us-east-1.s3.amazonaws.com/cicd/mf/pipeline.yaml>
  - <https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/pipeline.yaml>
- Seleziona Avanti.

 Note

Non modificare i valori dei parametri predefiniti a meno che non sia stato tu stesso a modificare il AWS CloudFormation modello.

4. In Configura le opzioni dello stack, scegli Avanti.
5. In Capabilities, scegli Riconosco che AWS CloudFormation potrebbe creare risorse IAM per consentire l'autorizzazione AWS CloudFormation a creare un ruolo IAM per tuo conto. Seleziona Crea stack.

 Note

Il provisioning di questo stack può richiedere dai 3 ai 5 minuti.

6. Dopo che lo stack è stato creato correttamente, vai alla sezione Output dello stack appena fornito. Qui troverai il bucket Amazon S3 in cui caricare il codice del mainframe e i file dipendenti.

Stack info	Events	Resources	Outputs	Parameters	Template	Change sets
<b>Outputs (7)</b>						
<input type="text" value="Search outputs"/>						
Key	Value	Description				
M2CICDNewPrivateSubnet1	subnet-0e1dda3ae86f025da	Subnet 1 for M2 CI/CD				
M2CICDNewPrivateSubnet2	subnet-0b89e607975284f8f	Subnet 2 for M2 CI/CD				
M2CICDNewVPC	vpc-034cbfc880b73dd28	VPC Id for M2 CI/CD				
MainframeCodeBucketS3URI	s3://mf-code-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Code S3 Bucket				
MainframeCodeBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Code S3 Bucket				
MainframeDataBucketS3URI	s3://mf-data-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Test Data S3 Bucket				
MainframeDataBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Test Data S3 Bucket				

## Crea un repository e una pipeline CI/CD AWS CodeCommit

In questo passaggio, crei un CodeCommit repository ed esegui il provisioning di uno stack di pipeline CI/CD chiamando una funzione Lambda che chiama per creare lo stack di pipeline. AWS CloudFormation

1. [Scaricate l'applicazione di esempio sul computer locale. BankDemo](#)
2. Carica `bankdemo.zip` dal tuo computer locale nel bucket Amazon S3 creato in [Crea l'infrastruttura di base della pipeline CI/CD](#)
3. Scarica `config_git.yml`.
4. Modifica il `config_git.yml` se necessario, come segue:
  - Aggiungi il nome del tuo repository di destinazione, il ramo di destinazione e il messaggio di commit.

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
```

- Aggiungi l'indirizzo email a cui desideri ricevere le notifiche.

```
pipeline-config:
  # Send pipeline failure notifications to these email addresses
  alert-notifications:
    - myname@mycompany.com
  # Send notifications for manual approval before production deployment to these
  email addresses
  approval-notifications:
    - myname@mycompany.com
```

5. Carica il `config_git.yml` file contenente la definizione della struttura delle cartelle del CodeCommit repository nel bucket Amazon S3 creato in [Crea l'infrastruttura di base della pipeline CI/CD](#) Ciò richiamerà la funzione Lambda che fornirà automaticamente il repository e la pipeline.

Questo creerà un CodeCommit repository con il nome fornito nel `config_git.yml` file `target-repository` definito, ad esempio. `bankdemo-repo`

La funzione Lambda creerà anche lo stack di pipeline CI/CD. AWS CloudFormation Lo AWS CloudFormation stack avrà lo stesso prefisso del `target-repository` nome fornito seguito da una stringa casuale (ad esempio. `bankdemo-repo-01234567` È possibile trovare l'URL del CodeCommit repository e l'URL per accedere alla pipeline creata nella console di gestione. AWS

**bankdemo-repo-mcdilnof** [Delete] [Update] [Stack actions ▼] [Create stack ▼]

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

**Outputs (2)** [Refresh] [Settings]

Search outputs

Key ▲	Value ▼	Description ▼
CodeCommitRepo	<a href="https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo">https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo</a>	HTTPS endpoint to clone the CodeCommit repository
PipelineURL	<a href="https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2">https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2</a>	URL to access the pipeline on AWS Management Console

6. Se la creazione del CodeCommit repository è completa, la pipeline CI/CD verrà attivata immediatamente per eseguire un CI/CD completo.

- Una volta che il file è stato inviato, attiverà automaticamente la pipeline che verrà creata, distribuita in staging, eseguita alcuni test e attenderà l'approvazione manuale prima di distribuirlo nell'ambiente di produzione.

## Esempio di file Trigger YAML config\_git.yml

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
  directory-structure:
    - '/':
      files:
        - build.xml
        - '*.yaml'
        - '*.yml'
        - '*.xml'
        - 'LICENSE.txt'
      readme: |
        # Root Folder
        - 'build.xml' : Build configuration for the application
    - tests:
      files:
        - '*.py'
      readme: |
        # Test Folder
        - '*.py' : Test scripts
    - config:
      files:
        - 'BANKDEMO.csd'
        - 'BANKDEMO.json'
        - 'BANKDEMO_ED.json'
        - 'dfhldrdat'
        - 'ESPGSQLXA.dll'
        - 'ESPGSQLXA64.so'
        - 'ESPGSQLXA64_S.so'
        - 'EXTFH.cfg'
        - 'm2-2021-04-28.normal.json'
        - 'MFDBFH.cfg'
        - 'application-definition-template-config.json'
      readme: |
        # Config Folder
```

This folder contains the application configuration files.

- 'BANKDEMO.csd' : CICS Resource definitions export file
- 'BANKDEMO.json' : Enterprise Server configuration
- 'BANKDEMO\_ED.json' : Enterprise Server configuration for ED
- 'dfhdrdat' : CICS resource definition file
- 'ESPGSQLXA.dll' : XA switch module Windows
- 'ESPGSQLXA64.so' : XA switch module Linux
- 'ESPGSQLXA64\_S.so' : XA switch module Linux
- 'EXTFH.cfg' : Micro Focus File Handler configuration
- 'm2-2021-04-28.normal.json' : M2 request document
- 'MFDBFH.cfg' : Micro Focus Database File Handler
- 'application-definition-template-config.json' : Application definition for

M2

- source:
  - subdirs:
    - .settings:
      - files:
        - '.bms.mfdirset'
        - '.cbl.mfdirset'
    - copybook:
      - files:
        - '\*.cpy'
        - '\*.inc'
      - readme: |
        - # Copy folder
        - This folder contains the source for COBOL copy books, PLI includes, ...
        - .cpy COBOL copybooks
        - .inc PLI includes
  - ctlcards:
    - files:
      - '\*.ctl'
      - 'KBNKSRT1.txt'
    - readme: |
      - # Control Card folder
      - This folder contains the source for Batch Control Cards
      - .ctl Control Cards
- ims:
  - files:
    - '\*.dbd'
    - '\*.psb'
  - readme: |
    - # ims folder
    - This folder contains the IMS DB source files with the extensions
    - .dbd for IMS DBD source

```
    - .psb for IMS PSB source
- jcl:
  files:
    - '*.jcl'
    - '*.ctl'
    - 'KBNKSRT1.txt'
    - '*.prc'
  readme: |
    # jcl folder
    This folder contains the JCL source files with the extensions
    - .jcl
#
# - proclib:
#   files:
#     - '*.prc'
#   readme: |
#     # proclib folder
#     This folder contains the JCL procedures referenced via PROCLIB
#     statements in the JCL with extensions
#     - .prc
- rdbms:
  files:
    - '*.sql'
  readme: |
    # rdbms folder
    This folder contains any DB2 related source files with extensions
    - .sql for any kind of SQL source
- screens:
  files:
    - '*.bms'
    - '*.mfs'
  readme: |
    # screens folder
    This folder contains the screens source files with the extensions
    - .bms for CICS BMS screens
    - .mfs for IMS MFS screens
  subdirs:
    - .settings:
      files:
        - '*.bms.mfdirset'
- cobol:
  files:
    - '*.cbl'
    - '*.pli'
  readme: |
```



```
    # source folder
    This folder contains the program source files with the extensions
    - .cbl for COBOL source
    - .pli for PLI source
  subdirs:
  - .settings:
      files:
        - '*.cbl.mfdirset'
- tests:
  files:
  - 'test_script.py'
  readme: |
    # tests Folder
    This folder contains the application test scripts
pipeline-config:
  alert-notifications:
  - myname@mycompany.com
  approval-notifications:
  - myname@mycompany.com
```

## Creazione di AppStream Enterprise Developer 2.0

Per configurare Micro Focus Enterprise Developer sulla AppStream versione 2.0, vedere [Tutorial: configurare Micro Focus Enterprise Developer sulla AppStream versione 2.0](#).

Per connettere il CodeCommit repository a Enterprise Developer, utilizzate il nome specificato `target-repository` in [Esempio di file Trigger YAML config\\_git.yml](#).

## Configurazione e test di Enterprise Developer

### Argomenti

- [Clonare il BankDemo CodeCommit repository in Enterprise Developer](#)
- [Crea un progetto COBOL BankDemo mainframe e crea un'applicazione](#)
- [Crea un ambiente BankDemo CICS e batch locale per i test](#)
- [Avvia il server BANKDEMO da Enterprise Developer](#)
- [Avvia il terminale Rumba 3270](#)
- [Esegui una BankDemo transazione](#)
- [Arresta il server BANKDEMO da Enterprise Developer](#)

Connect all'istanza Enterprise Developer AppStream 2.0 in cui è stata creata [Creazione di AppStream Enterprise Developer 2.0](#).

1. Avvia Enterprise Developer da Windows Start. Scegliete Micro Focus Enterprise Developer, quindi scegliete Enterprise Developer for Eclipse. Se state iniziando per la prima volta, potrebbe volerci del tempo.
2. In Eclipse Launcher, in Workspace: entra e **C:\Users\\workspace** scegli Avvia.

#### Note

Assicurati di scegliere la stessa posizione dopo la riconnessione all'istanza 2.0. AppStream La selezione dell'area di lavoro non è persistente.

3. In Benvenuto, scegliete Apri prospettiva COBOL. Questo verrà mostrato solo la prima volta per un nuovo spazio di lavoro.

## Clonare il BankDemo CodeCommit repository in Enterprise Developer

1. Scegli Finestra/Prospettiva/Apri prospettiva/Altro... /Git.
2. Scegli Clona un repository Git.
3. In Clone Git Repository, inserisci le seguenti informazioni:
  - In Location URI, inserisci l'URL HTTPS del CodeCommit repository.

#### Note

Copia l'URL del clone HTTPS per il CodeCommit repository nella console di AWS gestione e incollalo qui. L'URI verrà suddiviso nei percorsi Host e Repository.

- L'utente CodeCommit archivia le credenziali in Authentication User and Password e seleziona Archivia in Secure Store.
4. In Selezione del ramo, scegli Filiale principale, quindi scegli Avanti.
  5. In Destinazione locale, in Directory, inserisci **C:\Users\\workspace** e scegli Fine.

Il processo di clonazione è completato quando BANKDEMO [main] viene visualizzato nella vista Git Repositories.

## Crea un progetto COBOL BankDemo mainframe e crea un'applicazione

1. Passa alla prospettiva COBOL.
2. In Project, disattivate l'opzione Crea automaticamente.
3. In File, scegli Nuovo, quindi Progetto COBOL Mainframe.
4. In New Mainframe COBOL Project, inserite le seguenti informazioni:
  - In Nome progetto, immettete. BankDemo
  - Scegliete il modello Micro Focus [64 bit].
  - Scegli Fine.
5. In COBOL Explorer, espandete il nuovo BankDemo progetto.

### Note

[BANKDEMO main]tra parentesi quadre indica che il progetto è connesso al repository locale BankDemo CodeCommit .

6. Se la visualizzazione ad albero non mostra le voci relative ai programmi COBOL, ai copybook, ai sorgenti BMS e ai file JCL, scegliete Aggiorna dal menu contestuale del progetto. BankDemo
7. Dal menu BankDemo contestuale, scegliete Proprietà/Micro Focus/Impostazioni progetto/COBOL:
  - Scegliete Set di caratteri - ASCII.
  - Scegliete Applica, quindi Chiudi.
8. Se la compilazione dei sorgenti BMS e COBOL non viene avviata immediatamente, verificate nel menu Progetto che l'opzione Crea automaticamente sia abilitata.


L'output di Build verrà visualizzato nella vista Console e dovrebbe essere completato dopo alcuni minuti con messaggi BUILD SUCCESSFUL e. Build finished with no errors

L' BankDemo applicazione dovrebbe ora essere compilata e pronta per l'esecuzione locale.

## Crea un ambiente BankDemo CICS e batch locale per i test

1. In COBOL Explorer, espandi. BANKDEMO / config
2. Nell'editor, apriteBANKDEMO\_ED.json.

3. Trova la stringa ED\_Home= e modifica il percorso in modo che punti al progetto Enterprise Developer, come segue: `D:\\<username>\\workspace\\BANKDEMO`. Notate l'uso delle doppie barre (\\) nella definizione del percorso.
4. Salva e chiudi il file.
5. Scegli Server Explorer.
6. Dal menu contestuale predefinito, scegliete Apri pagina di amministrazione. La pagina di amministrazione di Micro Focus Enterprise Server viene aperta nel browser predefinito.
7. Solo per le sessioni AppStream 2.0, apportate le seguenti modifiche in modo da poter preservare la regione locale di Enterprise Server per i test locali:
  - In Directory Server/Default, scegliete PROPRIETÀ/Configurazione.
  - Sostituisci la posizione del repository con `D:\\<username>\\My Files\\Home Folder\\MFDS`.

 Note

È necessario completare i passaggi da 5 a 8 dopo ogni nuova connessione a un'istanza AppStream 2.0.

8. In Directory Server/Default, scegli Importa, quindi completa i seguenti passaggi:
  - Nel passaggio 1: Tipo di importazione, scegli JSON e scegli Avanti.
  - Nel Passaggio 2: Carica, fai clic per caricare il file nel quadrato blu.
  - In Scegli il file da caricare, inserisci:
    - Nome del file: `D:\\<username>\\workspace\\BANKDEMO\\config\\BANKDEMO_ED.json`.
    - Seleziona Apri.
  - Seleziona Avanti.
  - Nel passaggio 3: Le regioni cancellano le porte dagli endpoint.
  - Seleziona Avanti.
  - Nel passaggio 4: Importa, scegli Importa.
  - Scegli Fine.

L'elenco mostrerà ora un nuovo nome di server BANKDEMO.

## Avvia il server BANKDEMO da Enterprise Developer

1. Scegli Enterprise Developer.
2. In Server Explorer, scegli Predefinito, quindi scegli Aggiorna dal menu contestuale.

L'elenco dei server dovrebbe ora mostrare anche BANKDEMO.

3. Scegli BANKDEMO.
4. Dal menu contestuale, scegli Associa al progetto, quindi scegli BANKDEMO.
5. Dal menu contestuale, scegli Start.

La visualizzazione della console dovrebbe mostrare il registro per l'avvio del server.

Se BANKDEMO CASSI5030I PLTPI Phase 2 List(PI) Processing Completed viene visualizzato il messaggio, il Server è pronto per testare l'applicazione CICS BANKDEMO.

## Avvia il terminale Rumba 3270

1. Da Windows Start, avviate Micro Focus Rumba+ Desktop /Rumba+ Desktop.
2. In Benvenuto, scegliete CREATE NEW SESSION/Mainframe Display.
3. In Mainframe Display, scegliete Connessione/Configura.
4. In Configurazione della sessione, scegliete Connessione/TN3270.
5. In Nome host/Indirizzo, scegli Inserisci e inserisci l'indirizzo IP. 127.0.0.1
6. In Porta Telnet, inserisci la porta6000.
7. Scegli Applica.
8. Scegli Connetti.

La schermata di benvenuto CICS mostra una schermata con il messaggio di riga 1: This is the Micro Focus MFE CICS region BANKDEMO

9. Premi Ctrl+Shift+z per cancellare lo schermo.

## Esegui una BankDemo transazione

1. In una schermata vuota, inserisciBANK.
2. Nella schermata BANK10, nel campo di immissione dell'ID utente... :, inserisci guest e premi Invio.

3. Nella schermata BANK20, nel campo di immissione prima di Calcola il costo di un prestito, inserisci / (barra anteriore) e premi Invio.
4. Nella schermata BANK70:
  - Nella sezione L'importo che desideri prendere in prestito... :, entra10000.
  - In A un tasso di interesse di... :, inserisci5.0.
  - In Per quanti mesi... :, entra10.
  - Premere Invio.

Dovrebbe essere visualizzato il seguente risultato:

```
Resulting monthly payment.....: $1023.06
```

Questo completa la configurazione dell'applicazione BANKDEMO in Enterprise Developer.

## Arresta il server BANKDEMO da Enterprise Developer

1. In Server Explorer, scegli Predefinito, quindi scegli Aggiorna dal menu contestuale.
2. Scegli BANKDEMO.
3. Dal menu contestuale, scegli Stop.

La visualizzazione della console dovrebbe mostrare il registro relativo all'arresto del server.

Se `Server: BANKDEMO stopped successfully` viene visualizzato il messaggio, il server si è spento correttamente.

## Esercizio 1: Migliora il calcolo del prestito nell'applicazione BANKDEMO

### Argomenti

- [Aggiungi la regola di analisi del prestito a Enterprise Developer Code Analysis](#)
- [Fase 1: Eseguire l'analisi del codice per il calcolo del prestito](#)
- [Passaggio 2: modifica della mappa CICS BMS e del programma COBOL e test](#)
- [Fase 3: Aggiungere il calcolo dell'importo totale nel programma COBOL](#)
- [Fase 4: Confermare le modifiche ed eseguire la pipeline CI/CD](#)

In questo scenario, illustrerai il processo di modifica di esempio del codice, della sua implementazione e del suo test.

L'ufficio prestiti desidera un nuovo campo nella schermata di calcolo del prestito BANK70 per mostrare l'importo totale del prestito. Ciò richiede una modifica della schermata BMS MBANK70.CBL, l'aggiunta di un nuovo campo e del corrispondente programma di gestione dello schermo SBANK70P.CBL con i relativi quaderni. Inoltre, la routine di calcolo del prestito in BBANK70P.CBL deve essere estesa con la formula aggiuntiva.

Per completare questo esercizio, assicurati di soddisfare i seguenti prerequisiti.

- Scaricare [BANKDEMO-exercise.zip](#) in D:\PhotonUser\My Files\Home Folder
- Estrarre il file zip in D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise.
- Crea cartella D:\PhotonUser\My Files\Home Folder\AnalysisRules.
- Copia il file delle regole Loan+Calculation+Update.General-1.xml dalla BANKDEMO-exercise cartella in D:\PhotonUser\My Files\Home Folder\AnalysisRules.

#### Note

Le modifiche al codice in \*.CBL e \*.CPY sono contrassegnate con EXER01 nelle colonne da 1 a 6 per questo esercizio.

## Aggiungi la regola di analisi del prestito a Enterprise Developer Code Analysis

Le regole di analisi definite in Micro Focus Enterprise Analyzer possono essere esportate da Enterprise Analyzer e importate in Enterprise Developer per eseguire le stesse regole di analisi tra le fonti del progetto Enterprise Developer.

1. Aprire Window/Preferences/Micro Focus/COBOL/Code Analysis/Rules.
2. Scegli Modifica... e inserisci il nome della cartella D:\PhotonUser\My Files\Home Folder\AnalysisRules contenente il file delle regole Loan+Calculation+Update.General-1.xml.
3. Scegli Fine.
4. Scegli Applica, quindi scegli Chiudi.
5. Dal menu contestuale del progetto BANKDEMO, scegli Analisi del codice.

Dovresti vedere una voce per Loan Calculation Update.

## Fase 1: Eseguire l'analisi del codice per il calcolo del prestito

Con la nuova regola di analisi vogliamo identificare i programmi e le righe di codice COBOL presenti che corrispondono ai modelli di ricerca \*LOAN\* e \*RATE\* nelle espressioni \*PAYMENT\*, nelle istruzioni e nelle variabili. Questo ci aiuterà a navigare nel codice e a identificare le modifiche necessarie.

1. Dal menu contestuale del progetto BANKDEMO, scegli Analisi del codice/Aggiornamento del calcolo del prestito.

Questo eseguirà la regola di ricerca ed elencherà i risultati in una nuova scheda chiamata Code Analysis. L'esecuzione dell'analisi è completata quando la barra di avanzamento verde in basso a destra scompare.

La scheda Analisi del codice dovrebbe visualizzare un elenco esteso di BBANK20P.CBL, BBANK70P.CBL ciascuna delle quali elenca le istruzioni, le espressioni e le variabili che corrispondono ai modelli di ricerca. SBANK70P.CBL

Guardando il risultato, BBANK20P.CBL sono stati spostati solo i valori letterali che corrispondono al modello di ricerca. Quindi questo programma può essere ignorato.

2. Nella barra dei menu della scheda scegli - Icona per comprimere tutto.
3. Espandi SBANK70P.CBL e seleziona qualsiasi riga in qualsiasi ordine con un doppio clic per vedere come si aprirà il codice sorgente ed evidenzierà la riga selezionata nel codice sorgente. Riconoscerai anche che tutte le righe sorgente identificate sono contrassegnate.

## Passaggio 2: modifica della mappa CICS BMS e del programma COBOL e test

Per prima cosa cambieremo la mappa BMS, il programma SBANK70P.CBL di gestione dello schermo MBANK70.BMS e il quaderno per visualizzare il nuovo CBANKDAT.CPY campo. Per evitare di scrivere codice non necessaria in questo esercizio, nella cartella sono disponibili moduli sorgente modificati. D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01 Normalmente uno sviluppatore utilizza i risultati dell'analisi del codice per navigare e modificare i sorgenti. Se avete tempo e desiderate apportare le modifiche manuali, fatelo utilizzando le informazioni fornite in \*Modifica manuale in MBANK70.BMS e SBANK70P.CBL (opzionale)\*.



Per modifiche rapide, copia i seguenti file:

1. Da ..\BANKDEMO-exercise\Exercis01\screens\MBANK70.BMS a D:\PhotonUser\workspace\bankdemo\source\screens.
2. Da ..\BANKDEMO-exercise\Exercis01\cobol\SBANK70P.CBL a D:\PhotonUser\workspace\bankdemo\source\cobol.
3. Da ..\BANKDEMO-exercise\Exercis01\copybook\CBANKDAT.CPY a D:\PhotonUser\workspace\bankdemo\source\copybook.
4. Per assicurarti che tutti i programmi interessati dalle modifiche vengano compilati, scegli Project/Clean... /Pulisci tutto il progetto.

Per le modifiche manuali a MBANK70.BMS e SBANK70P.CBL, completa i seguenti passaggi:

- Per la modifica manuale del MBANK70.BMS codice sorgente BMS aggiungi dopo il PAYMENT campo:
  - TXT09 con gli stessi attributi di TXT08 e il valore INIZIALE «Importo totale del prestito»
  - TOTALE con gli stessi attributi di PAYMENT

Modifiche al test

Per testare le modifiche, ripeti i passaggi nelle seguenti sezioni:

1. [Avvia il server BANKDEMO da Enterprise Developer](#)
2. [Avvia il terminale Rumba 3270](#)
3. [Esegui una BankDemo transazione](#)

Inoltre ora dovresti vedere anche il testo Total Loan Amount.....:

4. [Arresta il server BANKDEMO da Enterprise Developer](#)

### Fase 3: Aggiungere il calcolo dell'importo totale nel programma COBOL

Nella seconda fase modificheremo SBANK70P.CBL e aggiungeremo il calcolo per l'importo totale del prestito. La fonte preparata con le modifiche richieste è disponibile nella D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01 cartella. Se hai tempo e

desideri apportare le modifiche manuali, fallo con le informazioni fornite in \*Modifica manuale in BBANK70P.CBL (opzionale) \*.

Per una modifica rapida, copia il seguente file:

- Da ..\BANKDEMO-exercise\Exercis01\source\cobol\BBANK70P.CBL a D:\PhotonUser\workspace\bankdemo\source\cobol.

Per apportare una modifica manuale aBBANK70P.CBL, completa i seguenti passaggi:

- Utilizzate il risultato dell'analisi del codice per identificare le modifiche richieste.

### Modifiche al test

Per testare le modifiche, ripeti i passaggi nelle seguenti sezioni:

1. [Avvia il server BANKDEMO da Enterprise Developer](#)
2. [Avvia il terminale Rumba 3270](#)
3. [Esegui una BankDemo transazione](#)

Inoltre ora dovresti vedere anche il testo `Total Loan Amount.....:`  
`$10230.60.`

4. [Arresta il server BANKDEMO da Enterprise Developer](#)

### Fase 4: Confermare le modifiche ed eseguire la pipeline CI/CD

Esegui il commit delle modifiche nell' CodeCommit archivio centrale e attiva la pipeline CI/CD per creare, testare e implementare le modifiche.

1. Dal progetto BANKDEMO, nel menu contestuale, scegli Team/Commit.
2. Nella scheda Git Staging, inserisci il seguente messaggio di commit:`Added Total Amount Calculation.`
3. Scegli Commit e Push... .
4. Apri la CodePipeline console e controlla lo stato dell'esecuzione della pipeline.

**Note**

In caso di problemi con la funzione Enterprise Developer o Teams Commit o Push, usa l'interfaccia a riga di comando Git Bash.

## Esercizio 2: Estrarre il calcolo del prestito nella domanda BankDemo

### Argomenti

- [Fase 1: rifattorizzare la routine di calcolo dei prestiti in una sezione COBOL](#)
- [Fase 2: Estrarre la routine di calcolo del prestito in un programma COBOL autonomo](#)
- [Fase 3: Conferma delle modifiche ed esecuzione della pipeline CI/CD](#)

In questo esercizio successivo, esaminerete un altro esempio di richiesta di modifica. In questo scenario, il reparto Prestiti desidera riutilizzare la routine di calcolo del prestito come procedura autonoma WebService. La routine dovrebbe rimanere in COBOL e dovrebbe inoltre essere ancora richiamabile dal programma CICS COBOL esistente. BBANK70P.CBL

### Fase 1: rifattorizzare la routine di calcolo dei prestiti in una sezione COBOL

Nella prima fase estraiamo la routine di calcolo del prestito in una sezione COBOL. Questo passaggio è necessario per estrarre il codice in un programma COBOL autonomo nel passaggio successivo.

1. Apri BBANK70P.CBL nell'editor COBOL.
2. Nell'editor, scegliete dal menu contestuale Code Analysis/Loan Calculation Update. Questo eseguirà la scansione della fonte corrente solo alla ricerca di modelli definiti nella regola di analisi.
3. Nel risultato nella scheda Analisi del codice, trova la prima istruzione aritmetica. DIVIDE WS-LOAN-INTEREST BY 12
4. Fai doppio clic sull'istruzione per accedere alla riga sorgente nell'Editor. Questa è la prima dichiarazione della routine di calcolo del prestito.
5. Contrassegna il seguente blocco di codice per la routine di calcolo del prestito da estrarre in una sezione.

```
DIVIDE WS-LOAN-INTEREST BY 12
```

```

        GIVING WS-LOAN-INTEREST ROUNDED.
        COMPUTE WS-LOAN-MONTHLY-PAYMENT ROUNDED =
            ((WS-LOAN-INTEREST * ((1 + WS-LOAN-INTEREST)
                ** WS-LOAN-TERM)) /
            (((1 + WS-LOAN-INTEREST) * WS-LOAN-TERM) - 1 ))
            * WS-LOAN-PRINCIPAL.
EXER01    COMPUTE WS-LOAN-TOTAL-PAYMENT =
EXER01        (WS-LOAN-MONTHLY-PAYMENT * WS-LOAN-TERM).

```

6. Dal menu contestuale dell'editor, scegli Refactor/Extract to Section... .
7. Inserisci il nome della nuova sezione: LOAN-CALCULATION.
8. Scegli OK.

Il blocco di codice contrassegnato è stato ora estratto LOAN-CALCULATION nella nuova sezione e il blocco di codice è stato sostituito con l'istruzione. PERFROM LOAN-CALCULATION

## Modifiche al test

Per testare le modifiche, ripeti i passaggi descritti nelle sezioni seguenti.

1. [Avvia il server BANKDEMO da Enterprise Developer](#)
2. [Avvia il terminale Rumba 3270](#)
3. [Esegui una BankDemo transazione](#)

Inoltre ora dovresti vedere anche il testo Total Loan Amount.....:  
\$10230.60.

4. [Arresta il server BANKDEMO da Enterprise Developer](#)

### Note

Se vuoi evitare i passaggi precedenti per estrarre il blocco di codice in una sezione, puoi copiare il codice sorgente modificato per il passaggio 1 da `..\BANKDEMO-exercise\Exercis02\Step1\cobol\BBANK70P.CBL` a `D:\PhotonUser\workspace\bankdemo\source\cobol`.

## Fase 2: Estrarre la routine di calcolo del prestito in un programma COBOL autonomo

Nel passaggio 2 il blocco di codice della LOAN-CALCULATION sezione verrà estratto in un programma autonomo e il codice originale verrà sostituito con il codice per richiamare il nuovo sottoprogramma.

1. Apri BBANK70P.CBL nell'editor e trova la nuova PERFORM LOAN-CALCULATION istruzione creata nel passaggio 1.
2. Posiziona il cursore all'interno del nome della sezione. Sarà contrassegnato in grigio.
3. Dal menu contestuale, seleziona Refactor->Estrai sezione/paragrafo dal programma... .
4. In Estrai sezione/paragrafo da programma, inserisci Nuovo nome di file: LOANCALC.CBL.
5. Scegli OK.

Il nuovo programma si aprirà nell'editor LOANCALC.CBL.

6. Scorri verso il basso e controlla il codice estratto e generato per l'interfaccia di chiamata.
7. Seleziona l'editor con BBANK70P.CBL e vai a. LOAN-CALCULATION SECTION Controlla il codice generato per chiamare il nuovo sottoprogramma LOANCALC.CBL.

### Note

L'CALListruzione utilizza DFHEIBLK e esegue chiamate LOANCALC con DFHCOMMAREA i blocchi di controllo CICS. Poiché vogliamo chiamare il nuovo LOANCALC.CBL sottoprogramma come programma non CICS, dobbiamo rimuovere DFHEIBLK e DFHCOMMAREA interrompere la chiamata commentando o eliminando.

### Modifiche ai test

Per testare le modifiche, ripeti i passaggi descritti nelle sezioni seguenti.

1. [Avvia il server BANKDEMO da Enterprise Developer](#)
2. [Avvia il terminale Rumba 3270](#)
3. [Esegui una BankDemo transazione](#)

Inoltre ora dovresti vedere anche il testo Total Loan Amount..... :  
\$10230.60.

4. [Arresta il server BANKDEMO da Enterprise Developer](#)

**Note**

Se vuoi evitare i passaggi precedenti per estrarre il blocco di codice in una sezione, puoi copiare il codice sorgente modificato per lo Step 1 da `..\BANKDEMO-exercise\Exercis02\Step2\cobo1\BBANK70P.CBL` e `LOANCALC.CBL` verso `D:\PhotonUser\workspace\bankdemo\source\cobo1`.

## Fase 3: Conferma delle modifiche ed esecuzione della pipeline CI/CD

Invia le modifiche al CodeCommit repository centrale e attiva la pipeline CI/CD per creare, testare e implementare le modifiche.

1. Dal progetto BANKDEMO, nel menu contestuale, scegli Team/Commit.
2. Nella scheda Git Staging
  - Aggiungi Unstaged Stages `LOANCALC.CBL` e `LoanCalc.cbl.MFDirSet`.
  - Inserisci un messaggio di commit: `Added Total Amount Calculation`
3. Scegli Commit e Push... .
4. Apri la CodePipeline console e controlla lo stato dell'esecuzione della pipeline.

**Note**

In caso di problemi con la funzione Enterprise Developer o Teams Commit o Push, usa l'interfaccia a riga di comando Git Bash.

## Pulizia delle risorse

Se non ti servono più le risorse che hai creato per questo tutorial, eliminale in modo da non continuare a ricevere costi aggiuntivi. Completa questa procedura:

- Elimina la CodePipeline pipeline. Per ulteriori informazioni, consulta [Eliminare una pipeline CodePipeline nella Guida](#) per l'AWS CodePipelineutente.
- Eliminare il CodeCommit repository. Per ulteriori informazioni, consulta [Eliminare un CodeCommit repository nella Guida](#) per l'AWS CodeCommitutente.

- Eliminare il bucket S3;. Per ulteriori informazioni, consulta [Eliminazione di un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
- Elimina lo stack AWS CloudFormation. Per ulteriori informazioni, vedi [Eliminazione di uno stack sulla console AWS CloudFormation](#) nella Guida per l'utente di AWS CloudFormation.

## Utilità Batch nella modernizzazione del AWS mainframe

Le applicazioni mainframe utilizzano spesso programmi di utilità batch per eseguire funzioni specifiche come l'ordinamento dei dati, il trasferimento di file tramite FTP, il caricamento di dati in database come DB2, lo scaricamento dei dati dai database e così via.

Quando si esegue la migrazione delle applicazioni a AWS Mainframe Modernization, sono necessarie utilità sostitutive funzionalmente equivalenti in grado di eseguire le stesse attività di quelle utilizzate sul mainframe. Alcune di queste utilità potrebbero essere già disponibili come parte dei motori di runtime di AWS Mainframe Modernization, ma stiamo fornendo le seguenti utilità sostitutive:

- M2SFTP: consente il trasferimento sicuro dei file tramite il protocollo SFTP.
- M2WAIT: attende un determinato periodo di tempo prima di continuare con il passaggio successivo di un processo batch.
- TXT2PDF: converte i file di testo in formato PDF.
- M2DFUTIL: fornisce funzioni di backup, ripristino, eliminazione e copia su set di dati simili al supporto fornito dall'utilità ADRDSSU del mainframe.
- M2RUNCMD: consente di eseguire comandi, script e chiamate di sistema Micro Focus direttamente da JCL.

Abbiamo sviluppato queste utilità batch sulla base del feedback dei clienti e le abbiamo progettate per fornire le stesse funzionalità delle utilità mainframe. L'obiettivo è rendere la transizione dalla modernizzazione del mainframe alla modernizzazione del mainframe il più AWS agevole possibile.

### Argomenti

- [Ubicazione binaria](#)
- [Utilità M2SFTP Batch](#)
- [Utilità M2WAIT Batch](#)
- [Utilità TXT2PDF Batch](#)
- [Utilità M2DFUTIL Batch](#)

- [Utilità M2RUNCMD Batch](#)

## Ubicazione binaria

Queste utilità sono preinstallate sui prodotti Micro Focus Enterprise Developer (ED) e Micro Focus Enterprise Server (ES). Puoi trovarle nella seguente posizione per tutte le varianti di ED ed ES:

- Linux: `/opt/aws/m2/microfocus/utilities/64bit`
- Windows (32 bit): `C:\AWS\M2\MicroFocus\Utilities\32bit`
- Windows (64 bit): `C:\AWS\M2\MicroFocus\Utilities\64bit`

## Utilità M2SFTP Batch

M2SFTP è un programma di utilità JCL progettato per eseguire trasferimenti sicuri di file tra sistemi che utilizzano il Secure File Transfer Protocol (SFTP). Il programma utilizza il client Putty SFTP per eseguire i trasferimenti di file `psftp` effettivi. Il programma funziona in modo simile a un programma di utilità FTP per mainframe e utilizza l'autenticazione di utenti e password.

### Note

L'autenticazione a chiave pubblica non è supportata.

Per convertire i JCL FTP del mainframe in modo che utilizzino SFTP, modificate in. `PGM=FTP`  
`PGM=M2SFTP`

### Argomenti

- [Piattaforme supportate](#)
- [Installazione delle dipendenze](#)
- [Configurare M2SFTP per Mainframe Modernization Managed AWS](#)
- [Configura M2SFTP per il runtime di modernizzazione AWS del mainframe su Amazon EC2 \(incluso 2.0\) AppStream](#)
- [JCL di esempio](#)
- [Riferimento ai comandi del client Putty SFTP \(PSFTP\)](#)
- [Passaggi successivi](#)



## Piattaforme supportate

È possibile utilizzare M2SFTP su una delle seguenti piattaforme:

- AWS Modernizzazione del mainframe | Micro Focus Managed
- Micro Focus Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Micro Focus Enterprise Developer (ED) e Micro Focus Enterprise Server (ES).

## Installazione delle dipendenze

Per installare il client SFTP Putty su Windows

- Scarica il client [SFTP PuTTY e](#) installalo.

Per installare il client SFTP Putty su Linux:

- Esegui il seguente comando per installare il client Putty SFTP:

```
sudo yum -y install putty
```

## Configurare M2SFTP per Mainframe Modernization Managed AWS

Se le applicazioni migrate sono in esecuzione su AWS Mainframe Modernization Managed, dovrai configurare M2SFTP come segue.

- Impostate le variabili di ambiente Micro Focus Enterprise Server appropriate per MFFTP. Ecco alcuni esempi:
  - MFFTP\_TEMP\_DIR
  - MFFTP\_SENDEOL
  - MFFTP\_TIME
  - MFFTP\_ABEND

Puoi impostare quante o quante di queste variabili desideri. Puoi impostarle nel tuo JCL usando l'ENVAR DDistribuzione. Per ulteriori informazioni su queste variabili, consultate [MFFTP Control Variables](#) nella documentazione di Micro Focus.

Per testare la configurazione, consulta. [JCL di esempio](#)

## Configura M2SFTP per il runtime di modernizzazione AWS del mainframe su Amazon EC2 (incluso 2.0) AppStream

Se le applicazioni migrate sono in esecuzione sul runtime di modernizzazione del AWS mainframe su Amazon EC2, configura M2SFTP come segue.

1. Modificate il [percorso del programma Micro Focus JES per includere la posizione binaria](#) per le utilità batch. Se è necessario specificare più percorsi, utilizzate i due punti (:) per separare i percorsi su Linux e il punto e virgola (;) su Windows. ;
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bit): C:\AWS\M2\MicroFocus\Utilities\64bit
2. Impostate le variabili di ambiente Micro Focus Enterprise Server appropriate per MFFTP. Ecco alcuni esempi:
  - MFFTP\_TEMP\_DIR
  - MFFTP\_SENDEOL
  - MFFTP\_TIME
  - MFFTP\_ABEND

È possibile impostare quante o quante di queste variabili si desidera. Puoi impostarle nel tuo JCL usando l'ENVAR DDistribuzione. Per ulteriori informazioni su queste variabili, consultate [MFFTP Control Variables](#) nella documentazione di Micro Focus.

Per testare la configurazione, consulta. [JCL di esempio](#)

## JCL di esempio

Per testare l'installazione, è possibile utilizzare uno dei seguenti file JCL di esempio.

### M2SFTP1.jcl

Questo JCL mostra come chiamare M2SFTP per inviare un file a un server SFTP remoto. Notate le variabili di ambiente impostate nell'istruzione. ENVVAR DD

```
//M2SFTP1 JOB 'M2SFTP1',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** Sample SFTP JCL step to send a file to SFTP server*
/**-----**
/**
//STEP01 EXEC PGM=M2SFTP,
//          PARM='127.0.0.1 (EXIT=99 TIMEOUT 300)'
/**
//SYSFTPD  DD  *
RECFM FB
LRECL 80
SBSENDEOL CRLF
MBSENDEOL CRLF
TRAILINGBLANKS FALSE
/*
//NETRC    DD  *
machine 127.0.0.1 login sftpuser password sftppass
/*
//SYSPRINT DD  SYSOUT=*
//OUTPUT  DD  SYSOUT=*
//STDOUT  DD  SYSOUT=*
//INPUT   DD  *
type a
locsite notrailingblanks
cd files
put 'AWS.M2.TXT2PDF1.PDF' AWS.M2.TXT2PDF1.pdf
put 'AWS.M2.CARDDEMO.CARDDATA.PS' AWS.M2.CARDDEMO.CARDDATA.PS1.txt
quit
/*
//ENVVAR   DD  *
```

```

MFFTP_VERBOSE_OUTPUT=ON
MFFTP_KEEP=N
/*
/**
//

```

## M2SFTP2.jcl

Questo JCL mostra come chiamare M2SFTP per ricevere un file da un server SFTP remoto. Notate le variabili di ambiente impostate nell'istruzione. ENVVAR DD

```

//M2SFTP2 JOB 'M2SFTP2',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** Sample SFTP JCL step to receive a file from SFTP server*
/**-----**
/**
//STEP01 EXEC PGM=M2SFTP
/**
//SYSPRINT DD SYSOUT=*
//OUTPUT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//INPUT DD *
open 127.0.0.1
sftpuser
sftppass
cd files
locsite recfm=fb lrecl=150
get AWS.M2.CARDDemo.CARDDATA.PS.txt +
'AWS.M2.CARDDemo.CARDDATA.PS2' (replace
quit
/*
//ENVVAR DD *
MFFTP_VERBOSE_OUTPUT=ON
MFFTP_KEEP=N
/*
/**
//

```

**Note**

Consigliamo vivamente di archiviare le credenziali FTP in un file NETRC e di limitare l'accesso ai soli utenti autorizzati.

## Riferimento ai comandi del client Putty SFTP (PSFTP)

Il client PSFTP non supporta tutti i comandi FTP. L'elenco seguente mostra tutti i comandi supportati da PSFTP.

Comando	Descrizione
!	Esegui un comando locale
addio	Termina la tua sessione SFTP
cd	Cambia la tua directory di lavoro remota
chmod	Modifica le autorizzazioni e le modalità dei file
close	Termina la sessione SFTP ma non uscire da PSFTP
del	Eliminare file sul server remoto
dir	Elenca i file remoti
Esci	Termina la sessione SFTP
get	Scarica un file dal server sul tuo computer locale
Aiuto	Offri aiuto
LCD	Cambia la directory di lavoro locale
lpwd	Stampa la cartella di lavoro locale
ls	Elenca i file remoti

Comando	Descrizione
mget	Scarica più file contemporaneamente
mkdir	Crea directory sul server remoto
Input	Carica più file contemporaneamente
mv	Sposta o rinomina file sul server remoto
aperto	Connect a un host
put	Carica un file dal tuo computer locale al server
pwd	Stampa la tua cartella di lavoro remota
Esci	Termina la sessione SFTP
rimpiango	Continua a scaricare i file
ren	Sposta o rinomina i file sul server remoto
riportare	Continua a caricare i file
rm	Eliminare file sul server remoto
rmdir	Rimuove le directory sul server remoto

## Passaggi successivi

Per caricare e scaricare file in Amazon Simple Storage Service utilizzando SFTP, puoi usare M2SFTP insieme a AWS Transfer Family, come descritto nei seguenti post di blog.

- [Utilizzo delle directory logiche AWS SFTP per creare un semplice servizio di distribuzione dei dati](#)
- [Abilita l'autenticazione tramite password per l'utilizzo AWS Transfer for SFTP AWS Secrets Manager](#)

## Utilità M2WAIT Batch

M2WAIT è un programma di utilità per mainframe che consente di introdurre un periodo di attesa negli script JCL specificando una durata in secondi, minuti o ore. Puoi chiamare M2WAIT direttamente da JCL passando il tempo che desideri attendere come parametro di input. Internamente, il programma M2WAIT richiama il modulo fornito da Micro Focus C\$SLEEP per attendere un periodo di tempo specificato.

### Note

È possibile utilizzare gli alias Micro Focus per sostituire ciò che è presente negli script JCL. Per ulteriori informazioni, vedete [JES Alias](#) nella documentazione di Micro Focus.

### Argomenti

- [Piattaforme supportate](#)
- [Configura M2WAIT for AWS Mainframe Modernization Managed](#)
- [Configura M2WAIT for AWS Mainframe Modernization runtime su Amazon EC2 \(incluso il 2.0\) AppStream](#)
- [Esempio JCL](#)

### Piattaforme supportate

Puoi utilizzare M2WAIT su una delle seguenti piattaforme:

- AWS Modernizzazione del mainframe | Micro Focus Managed
- Micro Focus Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Micro Focus Enterprise Developer (ED) e Micro Focus Enterprise Server (ES).

### Configura M2WAIT for AWS Mainframe Modernization Managed

Se le tue applicazioni migrate sono in esecuzione su AWS Mainframe Modernization Managed, dovrai configurare M2WAIT come segue.

- Usa il programma M2WAIT nel tuo JCL passando il parametro di input come mostrato in [Esempio JCL](#)

## Configura M2WAIT for AWS Mainframe Modernization runtime su Amazon EC2 (incluso il 2.0) AppStream

Se le applicazioni migrate sono in esecuzione sul runtime di modernizzazione del AWS mainframe su Amazon EC2, configura M2WAIT come segue.

1. Modificate il [percorso del programma Micro Focus JES per includere la posizione binaria](#) per le utilità batch. Se è necessario specificare più percorsi, utilizzate i due punti (:) per separare i percorsi su Linux e il punto e virgola (;) su Windows. ;
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bit): C:\AWS\M2\MicroFocus\Utilities\64bit
2. Usa il programma M2WAIT nel tuo JCL passando il parametro di input come mostrato in. [Esempio JCL](#)

### Esempio JCL

Per testare l'installazione, puoi usare il M2WAIT1.jcl programma.

Questo esempio di JCL mostra come chiamare M2WAIT e passarlo per diverse durate.

```
//M2WAIT1 JOB 'M2WAIT',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** Wait for 12 Seconds*
/**-----**
/**
//STEP01 EXEC PGM=M2WAIT,PARM='S012'
//SYSOUT DD SYSOUT=*
/**
/**-----**
/** Wait for 0 Seconds (defaulted to 10 Seconds)*
/**-----**
/**
//STEP02 EXEC PGM=M2WAIT,PARM='S000'
//SYSOUT DD SYSOUT=*
```



```
/**
/**-----**
/** Wait for 1 Minute*
/**-----**
/**
//STEP03 EXEC PGM=M2WAIT,PARM='M001'
//SYSOUT DD SYSOUT=*
/**
//
```

## Utilità TXT2PDF Batch

TXT2PDF è un programma di utilità mainframe comunemente usato per convertire un file di testo in un file PDF. Questa utilità utilizza lo stesso codice sorgente per TXT2PDF (z/OS freeware). L'abbiamo modificata per eseguirla nell'ambiente di runtime Micro Focus di Mainframe Modernization. AWS

### Argomenti

- [Piattaforme supportate](#)
- [Configura TXT2PDF per la modernizzazione gestita del mainframe AWS](#)
- [Configura TXT2PDF per il runtime di modernizzazione del AWS mainframe su Amazon EC2 \(inclusa la versione 2.0\) AppStream](#)
- [Esempio di JCL](#)
- [Modifiche](#)
- [Riferimenti](#)

### Piattaforme supportate

È possibile utilizzare TXT2PDF su una delle seguenti piattaforme:

- AWS Modernizzazione del mainframe | Micro Focus Managed
- Micro Focus Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Micro Focus Enterprise Developer (ED) e Micro Focus Enterprise Server (ES).

## Configura TXT2PDF per la modernizzazione gestita del mainframe AWS

Se le applicazioni migrate sono in esecuzione su AWS Mainframe Modernization Managed, configura TXT2PDF come segue.

- Create una libreria REXX EXEC chiamata. AWS.M2.REXX.EXEC Scaricate questi [moduli REXX](#) e copiateli nella libreria.
  - TXT2PDF.r<sub>ex</sub>- TXT2PDF z/OS freeware (modificato)
  - TXT2PDFD.r<sub>ex</sub>- TXT2PDF z/OS freeware (non modificato)
  - TXT2PDFX.r<sub>ex</sub>- TXT2PDF z/OS freeware (modificato)
  - M2GETOS.r<sub>ex</sub>- Per verificare il tipo di sistema operativo (Windows o Linux)

Per testare la configurazione, consulta [Esempio di JCL](#).

## Configura TXT2PDF per il runtime di modernizzazione del AWS mainframe su Amazon EC2 (inclusa la versione 2.0) AppStream

Se le applicazioni migrate sono in esecuzione sul runtime di modernizzazione del AWS mainframe su Amazon EC2, configura TXT2PDF come segue.

1. Imposta la variabile di ambiente Micro Focus sul valore appropriato, MFREXX\_CHARSET ad esempio «" per i dati ASCII. A

### Important

L'immissione di un valore errato potrebbe causare problemi di conversione dei dati (da EBCDIC ad ASCII), rendendo il PDF risultante illeggibile o inutilizzabile.

MFREXX\_CHARSETTi MF\_CHARSET consigliamo di impostarla in modo che corrisponda.

2. Modificate il [percorso del programma Micro Focus JES](#) per includere la posizione binaria per le utilità batch. Se è necessario specificare più percorsi, utilizzate i due punti (:) per separare i percorsi su Linux e il punto e virgola (;) su Windows. ;
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bit): C:\AWS\M2\MicroFocus\Utilities\64bit

3. Create una libreria REXX EXEC chiamata. AWS.M2.REXX.EXEC ` Scaricate questi [moduli REXX](#) e copiateli nella libreria.
- TXT2PDF.rex- TXT2PDF z/OS freeware (modificato)
  - TXT2PDFD.rex- TXT2PDF z/OS freeware (non modificato)
  - TXT2PDFX.rex- TXT2PDF z/OS freeware (modificato)
  - M2GETOS.rex- Per verificare il tipo di sistema operativo (Windows o Linux)

Per testare la configurazione, consulta [Esempio di JCL](#).

## Esempio di JCL

Per testare l'installazione, puoi utilizzare uno dei seguenti file JCL di esempio.

Txt2PDF1.jcl

Questo file JCL di esempio utilizza un nome DD per la conversione TXT2PDF.

```
//TXT2PDF1 JOB 'TXT2PDF1',CLASS=A,MSGCLASS=X,TIME=1440
//*
//* Copyright Amazon.com, Inc. or its affiliates.*
//* All Rights Reserved.*
//*
//*-----**
//* PRE DELETE*
//*-----**
//*
//PREDEL EXEC PGM=IEFBR14
//*
//DD01 DD DSN=AWS.M2.TXT2PDF1.PDF.VB,
// DISP=(MOD,DELETE,DELETE)
//*
//DD02 DD DSN=AWS.M2.TXT2PDF1.PDF,
// DISP=(MOD,DELETE,DELETE)
//*
//*-----**
//* CALL TXT2PDF TO CONVERT FROM TEXT TO PDF (VB)*
//*-----**
//*
//STEP01 EXEC PGM=IKJEFT1B
//*
//SYSEXEC DD DISP=SHR,DSN=AWS.M2.REXX.EXEC
```

```

/**
//INDD      DD *
1THIS IS THE FIRST LINE ON THE PAGE 1
0THIS IS THE THIRD LINE ON THE PAGE 1
-THIS IS THE 6TH LINE ON THE PAGE 1
THIS IS THE 7TH LINE ON THE PAGE 1
+_____ - OVERSTRIKE 7TH LINE
1THIS IS THE FIRST LINE ON THE PAGE 2
0THIS IS THE THIRD LINE ON THE PAGE 2
-THIS IS THE 6TH LINE ON THE PAGE 2
THIS IS THE 7TH LINE ON THE PAGE 2
+_____ - OVERSTRIKE 7TH LINE
/*
/**
//OUTDD     DD DSN=AWS.M2.TXT2PDF1.PDF.VB,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=256,DSORG=PS,RECFM=VB,BLKSIZE=0)
/**
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DDNAME=SYSIN
/**
//SYSIN    DD *
%TXT2PDF BROWSE Y IN DD:INDD +
OUT DD:OUTDD +
CC YES
/*
/**
/**-----**
/** CONVERT PDF (VB) TO PDF (LSEQ - BYTE STREAM)*
/**-----**
/**
//STEP02 EXEC PGM=VB2LSEQ
/**
//INFILE   DD DSN=AWS.M2.TXT2PDF1.PDF.VB,DISP=SHR
/**
//OUTFILE  DD DSN=AWS.M2.TXT2PDF1.PDF,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(LRECL=256,DSORG=PS,RECFM=LSEQ,BLKSIZE=0)
/**
//SYSOUT   DD SYSOUT=*
/**
//

```

## TXT2PDF2.jcl

Questo JCL di esempio utilizza un nome DSN per la conversione TXT2PDF.

```
//TXT2PDF2 JOB 'TXT2PDF2',CLASS=A,MSGCLASS=X,TIME=1440
//*
/* Copyright Amazon.com, Inc. or its affiliates.*
/* All Rights Reserved.*
/*
/*-----**
/* PRE DELETE*
/*-----**
/*
//PREDEL EXEC PGM=IEFBR14
/*
//DD01 DD DSN=AWS.M2.TXT2PDF2.PDF.VB,
// DISP=(MOD,DELETE,DELETE)
/*
//DD02 DD DSN=AWS.M2.TXT2PDF2.PDF,
// DISP=(MOD,DELETE,DELETE)
/*
/*-----**
/* CALL TXT2PDF TO CONVERT FROM TEXT TO PDF (VB)*
/*-----**
/*
//STEP01 EXEC PGM=IKJEFT1B
/*
//SYSEXEC DD DISP=SHR,DSN=AWS.M2.REXX.EXEC
/*
//INDD DD *
1THIS IS THE FIRST LINE ON THE PAGE 1
0THIS IS THE THIRD LINE ON THE PAGE 1
-THIS IS THE 6TH LINE ON THE PAGE 1
THIS IS THE 7TH LINE ON THE PAGE 1
+_____ - OVERSTRIKE 7TH LINE
1THIS IS THE FIRST LINE ON THE PAGE 2
0THIS IS THE THIRD LINE ON THE PAGE 2
-THIS IS THE 6TH LINE ON THE PAGE 2
THIS IS THE 7TH LINE ON THE PAGE 2
+_____ - OVERSTRIKE 7TH LINE
/*
/*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD DDNAME=SYSIN
```

```
/**
//SYSIN DD *
%TXT2PDF BROWSE Y IN DD:INDD +
OUT 'AWS.M2.TXT2PDF2.PDF.VB' +
CC YES
/*
/**
/**-----**
/** CONVERT PDF (VB) TO PDF (LSEQ - BYTE STREAM)*
/**-----**
/**
//STEP02 EXEC PGM=VB2LSEQ
/**
//INFILE DD DSN=AWS.M2.TXT2PDF2.PDF.VB,DISP=SHR
/**
//OUTFILE DD DSN=AWS.M2.TXT2PDF2.PDF,
// DISP=(NEW,CATLG,DELETE),
// DCB=(LRECL=256,DSORG=PS,RECFM=LSEQ,BLKSIZE=0)
/**
//SYSOUT DD SYSOUT=*
/**
//
```

## Modifiche

Per far funzionare il programma TXT2PDF nell'ambiente di runtime Micro Focus di AWS Mainframe Modernization, abbiamo apportato le seguenti modifiche:

- Modifiche al codice sorgente per garantire la compatibilità con il runtime Micro Focus REXX
- Modifiche per garantire che il programma possa essere eseguito su sistemi operativi Windows e Linux
- Modifiche per supportare sia il runtime EBCDIC che ASCII

## Riferimenti

Riferimenti TXT2PDF e codice sorgente:

- [Convertitore da testo a PDF](#)
- [z/OS Freeware TCP/IP e strumenti di posta](#)
- [Guida di riferimento per l'utente di TXT2PDF](#)

## Utilità M2DFUTIL Batch

M2DFUTIL è un programma di utilità JCL che fornisce funzioni di backup, ripristino, eliminazione e copia su set di dati, simili al supporto fornito dall'utilità ADRDSSU mainframe. Questo programma mantiene molti dei parametri SYSIN di ADRDSSU, il che semplifica il processo di migrazione a questa nuova utilità.

### Argomenti

- [Piattaforme supportate](#)
- [Requisiti della piattaforma](#)
- [Supporto futuro pianificato](#)
- [Ubicazioni degli asset](#)
- [Configura M2DFUTIL o il runtime di modernizzazione AWS del mainframe su Amazon EC2 \(incluso 2.0\) AppStream](#)
- [Sintassi generale](#)
- [JCL di esempio](#)

### Piattaforme supportate

È possibile utilizzare M2DFUTIL su una delle seguenti piattaforme:

- Micro Focus ES su Windows (64 bit e 32 bit)
- Micro Focus ES su Linux (64 bit)

### Requisiti della piattaforma

M2DFUTIL dipende dalla chiamata di uno script per eseguire un test di espressione regolare. In Windows, è necessario installare Windows Services for Linux (WSL) per eseguire questo script.

### Supporto futuro pianificato

Le funzionalità che attualmente non sono disponibili nell'utilità ADRDSSU per mainframe, ma che rientrano nell'ambito futuro includono:

- M2 Managed

- VSAM
- Supporto COPY per la ridenominazione dei nomi dei file
- Supporto RENAME per RESTORE
- INCLUDE ed ESCLUDE multipli
- Clausola BY per la sottoselezione tramite DSORG, CREDIT, EXPDT
- Clausola MWAIT per ripetere gli errori di accodamento
- Supporto di archiviazione S3 per DUMP/RESTORE

## Ubicazioni degli asset

Il modulo di caricamento di questa utilità viene chiamato M2DFUTIL . so su Linux e M2DFUTIL .dll su Windows. Questo modulo di caricamento si trova nelle seguenti posizioni:

- Linux: /opt/aws/m2/microfocus/utilities/64bit
- Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\32bit
- Windows (64 bit): C:\AWS\M2\MicroFocus\Utilities\64bit

Viene chiamato lo script utilizzato per il test delle espressioni regularicompate . sh. Questo script può essere trovato nelle seguenti posizioni:

- Linux: /opt/aws/m2/microfocus/utilities/scripts
- Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\scripts

## Configura M2DFUTIL o il runtime di modernizzazione AWS del mainframe su Amazon EC2 (incluso 2.0) AppStream

Configura la tua regione Enterprise Server con quanto segue:

- Aggiungere le seguenti variabili in [ES-Environment]
  - M2DFUTILS\_BASE\_LOC- La posizione predefinita per l'output DUMP
  - M2DFUTILS\_SCRIPTPATH- La posizione dello **compare .sh** script documentata in Asset Locations
  - M2DFUTILS\_VERBOSE- [VERBOSE o NORMAL]. Questo controlla il livello di dettaglio dell'output SYSPRINT



- Verificate che il percorso del modulo di carico sia aggiunto all'`JES\Configuration\JES Program Path` impostazione
- Verificate che gli script nella directory delle utilità dispongano delle autorizzazioni di esecuzione. È possibile aggiungere un'autorizzazione di esecuzione utilizzando il `chmod + x <script name>` comando, nell'ambiente Linux

## Sintassi generale

### DUMP

Offre la possibilità di copiare i file dall'attuale posizione catalogata a una posizione di backup. Attualmente questa posizione deve essere un file system.

#### Processo

DUMP eseguirà le seguenti operazioni:

1. Crea la directory della posizione di destinazione.
2. Cataloga la directory delle posizioni di destinazione come membro PDS.
3. Determinate i file da includere elaborando il parametro INCLUDE.
4. Deseleziona i file inclusi elaborando il parametro EXCLUDE.
5. Determina se i file scaricati devono essere ELIMINATI.
6. Metti in coda i file da elaborare.
7. Copia i file.
8. Esporta le informazioni DCB catalogate dei file copiati in un file laterale nella posizione di destinazione per facilitare le future operazioni di RESTORE.

#### Sintassi

```
DUMP
TARGET ( TARGET LOCATION )      -
INCLUDE ( DSN. )
[ EXCLUDE ( DSN ) ]
[ CANCEL | IGNORE ]
[ DELETE ]
```

## Parametri obbligatori

Di seguito sono riportati i parametri richiesti per DUMP:

- SYSPRINT DD NAME- Per contenere informazioni di registrazione aggiuntive
- TARGET- Ubicazione dell'obiettivo. Può essere:
  - Percorso completo della posizione di scarico
  - Nome della sottodirectory creato nella posizione definita nella variabile M2DFUTILS\_BASE\_LOC
- INCLUDE- Un singolo nome DSN o una stringa di ricerca DSN mainframe valida
- EXCLUDE- Un solo nome DSN o una stringa di ricerca DSN mainframe valida

## Parametri facoltativi

- ANNULLA - Annulla se si verifica un errore. I file che sono stati elaborati verranno conservati
- (Impostazione predefinita) IGNORE: ignora qualsiasi errore e processo fino alla fine
- ELIMINA: se non si verifica alcun errore ENQ, il file viene eliminato e non viene catalogato

## DELETE

Offre la possibilità di eliminare e annullare la catalogazione in massa dei file. Non viene eseguito il backup dei file.

## Processo

DELETE eseguirà le seguenti operazioni:

1. Determina i file da includere elaborando il parametro INCLUDE.
2. Deseleziona i file inclusi elaborando il parametro EXCLUDE.
3. Accoda i file da elaborare. Impostazione della disposizione su OLD, DELETE, KEEP.

## Sintassi

```
DELETE  
INCLUDE ( DSN )  
[ EXCLUDE ( DSN ) ]  
[ CANCEL | IGNORE ]  
[ DELETE ]
```

## Parametri obbligatori

Di seguito sono riportati i parametri richiesti per DELETE:

- SYSPRINT DD NAME- Per contenere informazioni di registrazione aggiuntive
- INCLUDE- Un singolo nome DSNNAME o una stringa di ricerca DSN mainframe valida
- EXCLUDE- Un solo nome DSNNAME o una stringa di ricerca DSN mainframe valida

## Parametri facoltativi

- ANNULLA - Annulla se si verifica un errore. I file elaborati verranno conservati
- (Impostazione predefinita) IGNORE: ignora qualsiasi errore e processo fino alla fine

## RESTORE

Offre la possibilità di ripristinare i file di cui è stato precedentemente eseguito il backup utilizzando DUMP. I file vengono ripristinati nella posizione originale catalogata a meno che non venga utilizzato RENAME per modificare il DSNNAME ripristinato.

## Processo

RESTORE eseguirà le seguenti operazioni:

1. Convalida la directory di posizione di origine.
2. Determinate i file da includere elaborando il file di esportazione del catalogo.
3. Deselezionate i file inclusi elaborando il parametro EXCLUDE.
4. Accoda i file da elaborare.
5. File di catalogo che non sono catalogati in base alle informazioni di esportazione.
6. Se un file è già catalogato e le informazioni del catalogo di esportazione sono le stesse, RESTORE sostituirà il set di dati catalogato se è impostata l'opzione REPLACE.

## Sintassi

```
RESTORE  
SOURCE ( TARGET LOCATION )  
INCLUDE ( DSN )  
[ EXCLUDE ( DSN ) ]
```

```
[ CANCEL | IGNORE ]  
[ REPLACE]
```

## Parametri obbligatori

Di seguito sono riportati i parametri richiesti per RESTORE:

- SYSPRINT DD NAME- Per contenere informazioni di registrazione aggiuntive
- SOURCE- Ubicazione della fonte. Può essere:
  - Percorso completo della posizione di scarico
  - Nome della sottodirectory creato nella posizione definita nella variabile M2DFUTILS\_BASE\_LOC
- INCLUDE- Un singolo nome DSNAME o una stringa di ricerca DSN mainframe valida
- EXCLUDE- Un solo nome DSNAME o una stringa di ricerca DSN mainframe valida

## Parametri facoltativi

- ANNULLA - Annulla in caso di errore. I file elaborati vengono conservati
- (Impostazione predefinita) IGNORE: ignora qualsiasi errore e processo fino alla fine
- SOSTITUISCI: se il file da ripristinare è già catalogato e i record del catalogo sono gli stessi, sostituisci il file catalogato

## JCL di esempio

### Lavoro DUMP

Questo lavoro creerà una sottodirectory chiamata. TESTDUMP Questa è la posizione di backup predefinita specificata dalla variabile M2DFUTILS\_BASE\_LOC. Creerà una libreria PDS per questo backup chiamata. M2DFUTILS .TESTDUMP I dati del catalogo esportati vengono memorizzati in un file sequenziale di righe nella directory di backup denominata. CATDUMP .DAT Tutti i file selezionati verranno copiati in questa directory di backup.

```
//M2DFDMP JOB 'M2DFDMP',CLASS=A,MSGCLASS=X  
//STEP001 EXEC PGM=M2DFUTIL  
//SYSPRINT DD DSN=TESTDUMP.SYSPRINT,  
//          DISP=(NEW,CATLG,DELETE),  
//          DCB=(RECFM=LSEQ,LRECL=256)  
//SYSIN    DD *
```

```

DUMP TARGET(TESTDUMP)          -
      INCLUDE(TEST.FB.FILE*.ABC) -
CANCEL
/*
//

```

## CANCELLA il lavoro

Questo processo eliminerà tutti i file dal catalogo che corrispondono al parametro INCLUDE.

```

/M2DFDEL JOB 'M2DFDEL',CLASS=A,MSGCLASS=X
//STEP001 EXEC PGM=M2DFUTIL
//SYSPRINT DD DSN=TESTDEL.SYSPRINT,
//      DISP=(NEW,CATLG,DELETE),
//      DCB=(RECFM=LSEQ,LRECL=256)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
DELETE                          -
      INCLUDE(TEST.FB.FILE*.ABC) -
CANCEL
/*
//

```

## Processo RESTORE

Questo processo ripristinerà i file che corrispondono al parametro INCLUDE dalla posizione TESTDUMP di backup. I file catalogati verranno sostituiti se il file catalogato è uguale a quello nell'esportazione CATDUMP e viene specificata l'opzione REPLACE.

```

//M2DFREST JOB 'M2DFREST',CLASS=A,MSGCLASS=X
//STEP001 EXEC PGM=M2DFUTIL
////SYSPRINT DD DSN=TESTREST.SYSPRINT,
//      DISP=(NEW,CATLG,DELETE),
//      DCB=(RECFM=LSEQ,LRECL=256)
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
RESTORE SOURCE(TESTDUMP)        -
      INCLUDE(TEST.FB.FILE*.ABC) -
IGNORE
REPLACE
/*
//

```

## Utilità M2RUNCMD Batch

È possibile utilizzare M2RUNCMD, un programma di utilità batch, per eseguire comandi, script e chiamate di sistema Micro Focus direttamente da JCL anziché eseguirli da un terminale o da un prompt dei comandi. L'output dei comandi viene registrato nel registro dello spool del processo batch.

### Argomenti

- [Piattaforme supportate](#)
- [Configura M2RUNCMD per il runtime di modernizzazione AWS del mainframe su Amazon EC2 \(incluso 2.0\) AppStream](#)
- [JCL di esempio](#)

### Piattaforme supportate

È possibile utilizzare M2RUNCMD sulle seguenti piattaforme:

- Micro Focus Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Micro Focus Enterprise Developer (ED) e Micro Focus Enterprise Server (ES).

### Configura M2RUNCMD per il runtime di modernizzazione AWS del mainframe su Amazon EC2 (incluso 2.0) AppStream

Se le applicazioni migrate sono in esecuzione sul runtime di modernizzazione del AWS mainframe su Amazon EC2, configura M2RUNCMD come segue.

- Modificate il [percorso del programma Micro Focus JES](#) per includere la posizione binaria per le utilità batch. Se è necessario specificare più percorsi, utilizzate i due punti (:) per separare i percorsi su Linux e il punto e virgola (;) su Windows.
  - Linux: /opt/aws/m2/microfocus/utilities/64bit
  - Windows (32 bit): C:\AWS\M2\MicroFocus\Utilities\32bit
  - Windows (64 bit): C:\AWS\M2\MicroFocus\Utilities\64bit

### JCL di esempio

Per testare l'installazione, è possibile utilizzare uno dei seguenti JCL di esempio.

## Esegui SCRL1.jcl

Questo JCL di esempio crea uno script e lo esegue. Il primo passaggio crea uno script chiamato /tmp/TEST\_SCRIPT.sh e con contenuti provenienti da dati SYSUT1 in-stream. Il secondo passaggio imposta l'autorizzazione di esecuzione ed esegue lo script creato nel primo passaggio. Potete anche scegliere di eseguire solo il secondo passaggio per eseguire i comandi Micro Focus e di sistema già esistenti.

```
//RUNSCRL1 JOB 'RUN SCRIPT',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
/*-----*
/*  CREATE SCRIPT (LINUX)
/*-----*
//*
//STEP0010 EXEC PGM=IEBGENER
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//SYSUT1   DD *
#!/bin/bash

set -x

## ECHO PATH ENVIRONMENT VARIABLE
echo $PATH

## CLOSE/DISABLE VSAM FILE
casfile -r$ES_SERVER -oc -ed -dACCTFIL

## OPEN/ENABLE VSAM FILE
casfile -r$ES_SERVER -ooi -ee -dACCTFIL

exit $?
/*
//SYSUT2   DD DSN=&&TEMP,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=LSEQ,LRECL=300,DSORG=PS,BLKSIZE=0)
/*MFE: %PCDSN='/tmp/TEST_SCRIPT.sh'
/*
/*-----*
/*  RUN SCRIPT (LINUX)
/*-----*
```

```
/**-----*  
/**  
//STEP0020 EXEC PGM=RUNCMD  
/**  
//SYSOUT DD SYSOUT=*  
/**  
//SYSIN DD *  
*RUN SCRIPT  
  sh /tmp/TEST_SCRIPT.sh  
/*  
//
```

## SYSOUT

L'output del comando o dello script che viene eseguito viene scritto nel SYSOUT registro. Per ogni comando eseguito, visualizza il comando, l'output e il codice restituito.

```
***** CMD Start *****  
  
CMD_STR: sh /tmp/TEST_SCRIPT.sh  
  
CMD_OUT:  
  
+ echo /opt/microfocus/EnterpriseServer/bin:/sbin:/bin:/usr/sbin:/usr/bin  
/opt/microfocus/EnterpriseServer/bin:/sbin:/bin:/usr/sbin:/usr/bin  
+ casfile -rMYDEV -oc -ed -dACCTFIL  
  
-Return Code: 0  
  
Highest return code: 0  
  
+ casfile -rMYDEV -ooi -ee -dACCTFIL  
  
-Return Code: 8  
  
Highest return code: 8  
  
+ exit 8  
  
CMD_RC=8
```



```
*****      CMD End      *****
```

## Esegui cmd11.jcl

Questo JCL di esempio utilizza RUNCMD per eseguire più comandi.

```
//RUNCMDL1 JOB 'RUN CMD',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
//*-----*
//*  RUN SYSTEM COMMANDS                                *
//*-----*
//*
//STEP0001 EXEC PGM=RUNCMD
//*
//SYSOUT DD SYSOUT=*
//*
//SYSIN DD *
*LIST DIRECTORY
  ls
*ECHO PATH ENVIRONMENT VARIABLE
  echo $PATH
/*
//
```

# AWS Modernizzazione del mainframe e replica dei dati con Precisly

AWS La modernizzazione del mainframe offre una varietà di Amazon Machine Images (AMI). Queste AMI facilitano il provisioning rapido delle istanze Amazon EC2, creando un ambiente personalizzato per la replica dei dati dai sistemi mainframe all'utilizzo di Precisly. AWS Questa guida fornisce i passaggi necessari per accedere e utilizzare queste AMI.

## Prerequisiti

- Assicurati di avere accesso da amministratore a un AWS account in cui puoi creare istanze Amazon EC2.
- Verifica che il servizio di modernizzazione del AWS mainframe sia disponibile nella regione in cui intendi creare le istanze Amazon EC2. Consulta [l'elenco dei servizi AWS disponibili per regione](#).
- Identifica l'Amazon Virtual Private Cloud (Amazon VPC) in cui verranno create le istanze Amazon EC2.
- Quando crei istanze Amazon EC2 in un Amazon VPC, assicurati che la tabella di routing associata disponga di un gateway Internet o di un gateway NAT.

### Note

Una replica dei dati di successo richiede che l'istanza AWS EC2 abbia accesso alla comunicazione con AWS Marketplace. In caso di problemi di connettività con AWS Marketplace, il processo di replica avrà esito negativo.

## Iscriviti all'Amazon Machine Image

Quando ti abboni a un prodotto AWS Marketplace, puoi avviare un'istanza dall'AMI del prodotto.

1. Accedi AWS Management Console e apri la Marketplace AWS console all'[indirizzo https://console.aws.amazon.com/marketplace](https://console.aws.amazon.com/marketplace).
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).

3. Copia e incolla il seguente link nella barra degli indirizzi del browser: <https://aws.amazon.com/marketplace/pp/prodview-en3xrbgzbs3dk>
4. Scegliere Continue to Subscribe (Continua la sottoscrizione).
5. Se i termini e le condizioni sono accettabili, scegli Accetta termini. L'elaborazione dell'abbonamento potrebbe richiedere alcuni minuti.
6. Attendi che venga visualizzato il messaggio di ringraziamento, come mostrato di seguito. Questo messaggio conferma che l'iscrizione al prodotto è avvenuta con successo.



## AWS Mainframe Modernization service Data Replication with Precisely

Thank you for subscribing to this product! You can now configure your software.

7. Nel riquadro di navigazione a sinistra, scegli Gestisci abbonamenti. Questa visualizzazione mostra tutti gli abbonamenti a cui sei abbonato.

## Avvia la replica dei dati di AWS Mainframe Modernization con Precisly

1. [Apri la Marketplace AWS console all'indirizzo https://console.aws.amazon.com/marketplace.](https://console.aws.amazon.com/marketplace)
2. Nel riquadro di navigazione a sinistra, scegli Gestisci abbonamenti.
3. Trova l'AMI che desideri avviare e scegli Launch new instance.
4. In Regione, seleziona la regione consentita.
5. Scegli Continua a lanciare tramite EC2. Questa azione ti porta alla console Amazon EC2.
6. Inserisci un nome per il server.
7. Seleziona un tipo di istanza che soddisfi i requisiti di prestazioni e costi del progetto. Il punto di partenza consigliato per la dimensione dell'istanza è c5.2xLarge.
8. Scegli una key pair esistente o creane e salvane una nuova. Per informazioni sulle coppie di chiavi, consulta le coppie di [chiavi Amazon EC2 e le istanze Linux](#) nella Amazon EC2 User Guide for Linux Instances.
9. Modifica le impostazioni di rete e scegli il VPC nell'elenco consentito e la sottorete appropriata.

10. Scegli un gruppo di sicurezza esistente o creane uno nuovo. Oltre a consentire l'accesso SSH (per impostazione predefinita sulla porta 22), per la replica dei dati con un'istanza EC2 del server Precisly, è tipico consentire il traffico TCP alla porta predefinita 2626.
11. Configura lo storage per l'istanza Amazon EC2.
12. Controlla il riepilogo e scegli Launch instance. Affinché il lancio abbia successo, il tipo di istanza deve essere valido. Se l'avvio fallisce, scegli Modifica configurazione dell'istanza e scegli un tipo di istanza diverso.
13. Dopo aver visualizzato il messaggio di successo, scegli Connect to instance.
14. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
15. Nel riquadro di navigazione a sinistra, nel menu Istanze, scegli Istanze.
16. Nel riquadro principale, controlla lo stato della tua istanza.

## Creazione di una policy IAM

Per utilizzare correttamente le istanze EC2 di AWS Mainframe Modernization distribuite tramite il nostro Marketplace AWS elenco, devi configurare un ruolo e una policy IAM. Questa configurazione IAM specificamente personalizzata non è facoltativa; autorizza le istanze Amazon EC2 a interagire con il servizio. Marketplace AWS Il ruolo e la policy di IAM consentono a AWS Mainframe Modernization di registrare con precisione i dati di utilizzo, essenziali per una fatturazione precisa. La mancata implementazione di questa configurazione può portare a tentativi falliti di replica dei dati e interruzioni operative.

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).
3. Se è la prima volta che scegli Policies, viene visualizzata la pagina Benvenuto in Managed Policies. Seleziona Inizia.
4. Nella parte superiore della pagina, scegli Crea policy.
5. Nella sezione Editor di policy, scegli l'opzione JSON.
6. Inserisci la seguente politica JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["aws-marketplace:MeterUsage"],
```

```
        "Effect": "Allow",
        "Resource": "*"
    }
]
}
```

## Creazione di un ruolo IAM

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 Tipo di entità attendibile, scegli Servizio AWS .
4. Nella sezione Caso d'uso, in Servizio o caso d'uso, scegli Amazon EC2.
5. Seleziona Avanti.
6. Nell'elenco delle politiche, seleziona Customer managed dal menu a discesa Filtra per tipo e inserisci il nome della policy che hai creato. Seleziona la casella di controllo accanto al nome della politica.
7. Seleziona Avanti.
8. Inserisci un nome e, facoltativamente, una descrizione per il ruolo.
9. Esamina la politica di fiducia e le autorizzazioni, quindi scegli Crea ruolo.

## Collega il ruolo IAM all'istanza Amazon EC2

1. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, seleziona Istanze.
3. Seleziona la tua istanza Amazon EC2.
4. Dal menu Azioni, scegli Sicurezza, quindi scegli Modifica ruolo IAM.
5. Seleziona il ruolo da collegare all'istanza, quindi scegli Aggiorna ruolo IAM.

# Integrazione con Charon

## Introduzione a Charon-SSP

Nel 1987, Sun Microsystems ha rilasciato il processore SPARC V7, un processore RISC a 32 bit. Nel 1990 seguì lo SPARC V8, una revisione dell'originale SPARC V7, con l'inclusione più importante delle istruzioni hardware di divisione e moltiplicazione. I processori SPARC V8 costituivano la base per una serie di server e workstation come le SPARCstation 5, 10 e 20. Nel 1993, lo SPARC V8 è stato seguito dal processore SPARC V9 a 64 bit. Anche questo divenne la base per una serie di server e workstation, come Enterprise 250 e 450.

A causa dell'obsolescenza dell'hardware e della mancanza di parti di ricambio o ricondizionate, il software e i sistemi sviluppati per queste vecchie workstation e server basati su SPARC sono diventati più difficili da mantenere. Per soddisfare la continua esigenza di determinati sistemi end-of-life basati su SPARC, Stromasys S.A. ha sviluppato la linea Charon-SSP di prodotti di emulazione SPARC. I seguenti prodotti sostituiscono macchine virtuali basate su software per i sistemi SPARC con hardware nativo specificati. Di seguito è riportata una panoramica generale delle famiglie di hardware emulate.

Charon-SSP/4M emula il seguente hardware SPARC:

- Famiglia Sun-4m (rappresentata dalla Sun SPARCstation 20): originariamente, una variante multiprocessore Sun-4, basata sul bus del modulo processore MBus introdotto nella serie SPARCServer 600MP. L'architettura Sun-4m in seguito comprendeva anche sistemi uniprocessore non MBus come SPARCstation 5, che utilizzavano processori con architettura SPARC V8. Supportato a partire da SunOS 4.1.2 e da Solaris 2.1 a Solaris 9. Il supporto per SPARCServer 600MP è stato interrotto dopo Solaris 2.5.1.

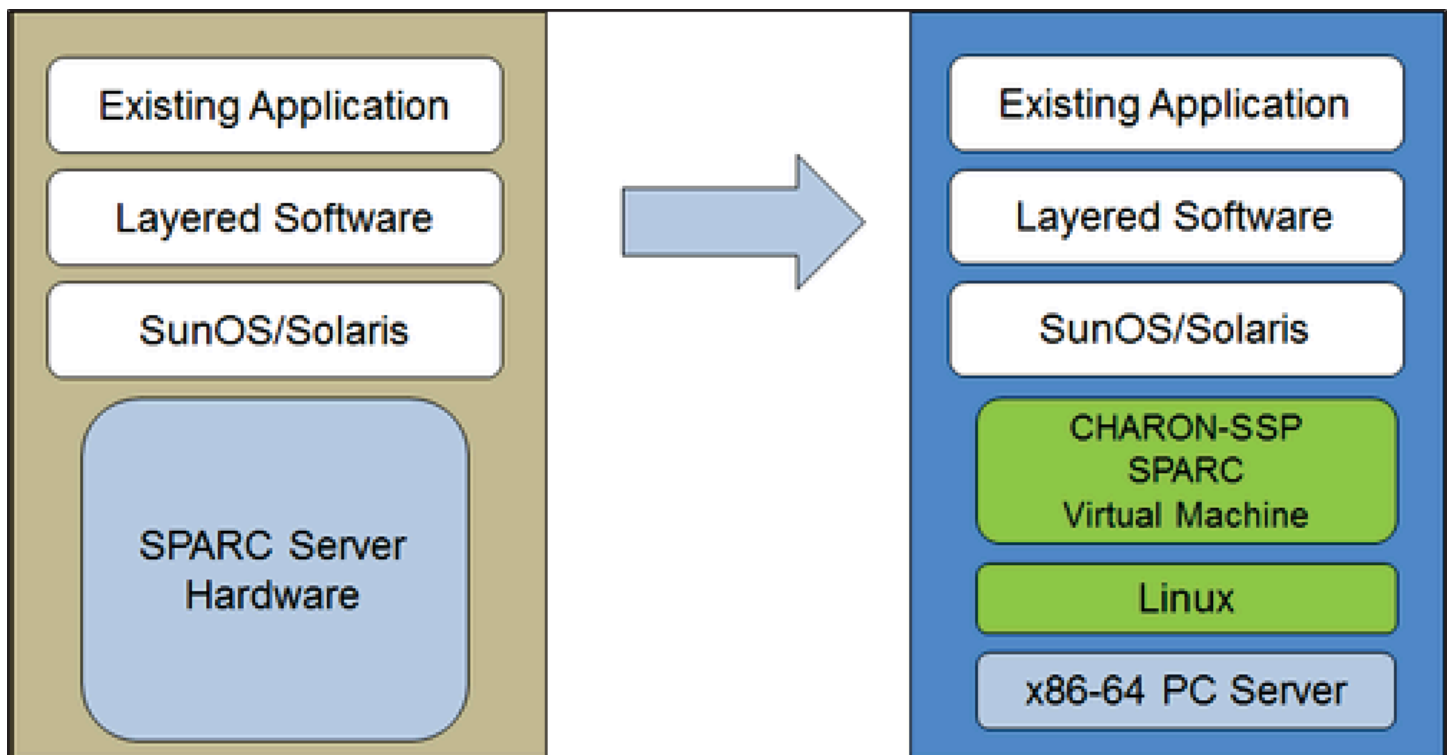
Charon-SSP/4U (+) emula il seguente hardware SPARC:

- Famiglia Sun-4u (rappresentata da Sun Enterprise 450): (U per UltraSPARC): questa variante ha introdotto l'architettura del processore SPARC V9 a 64 bit e l'interconnessione del processore UPA, utilizzate per la prima volta nella serie Sun Ultra. Supportata dalle versioni a 32 bit di Solaris a partire dalla versione 2.5.1. La prima versione di Solaris a 64 bit per Sun-4u è stata Solaris 7. Il supporto per UltraSPARC I è stato interrotto dopo Solaris 9. Solaris 10 supporta le implementazioni Sun-4u da UltraSPARC II a UltraSPARC IV.

Charon-SSP/4v (+) emula il seguente hardware SPARC:

- Famiglia Sun-4v (rappresentata dai modelli SPARC T2 e T4): questa variante ha aggiunto la virtualizzazione del processore hypervisor al Sun-4U, introdotta nel processore multicore Ultra SPARC T1. L'hardware selezionato era supportato da Solaris 10 a partire dalla versione 3/05 HW2 (la maggior parte dei modelli, incluso l'hardware emulato da Charon-SSP, richiede versioni più recenti di Solaris 10). Sono supportate anche diverse versioni di Solaris 11.

L'immagine seguente mostra il concetto di base della migrazione dell'hardware fisico verso un emulatore.



Le macchine virtuali Charon-SSP consentono agli utenti di computer basati su Sun e Oracle SPARC di sostituire l'hardware nativo in un modo che richiede modifiche minime o nulle alla configurazione originale del sistema. Ciò significa che è possibile continuare a eseguire le applicazioni e i dati senza la necessità di passare a un'altra piattaforma o trasferirli su un'altra piattaforma. Il software Charon-SSP funziona su sistemi standard Intel a 64 bit, garantendo la protezione continua del vostro investimento.

Charon-SSP/4U+ supporta le stesse piattaforme SPARC virtuali di Charon-SSP/4U e Charon-SSP/4V+ le stesse di Charon-SSP/4V. Tuttavia, le versioni 4U+ e 4V+ sfruttano la tecnologia di virtualizzazione assistita da hardware VTX/EPT di Intel e AMD-v/NPT di AMD nelle moderne CPU

per offrire migliori prestazioni della CPU virtuale. Charon-SSP/4U+ e Charon-SSP/4V+ richiedono CPU con supporto VT-X/EPT o AMD-V/NPT e devono essere installate su un sistema host dedicato. L'esecuzione di queste varianti di prodotto in una macchina virtuale (ad esempio, su VMware) non è supportata.

#### Note

Se prevedi di eseguire Charon-SSP/4U+ o 4V+ in un ambiente cloud, contatta Stromasys o un VAR Stromasys per discutere delle tue esigenze.

## Sistemi operativi guest supportati

Le macchine virtuali Charon-SSP/4M supportano le seguenti versioni del sistema operativo guest:

- SunOS 4.1.3 - 4.1.4
- Da Solaris 2.3 a Solaris 9

Le macchine virtuali Charon-SSP/4U (+) supportano le seguenti versioni del sistema operativo guest:

- Da Solaris 2.5.1 a Solaris 10

Le macchine virtuali Charon-SSP/4v (+) supportano le seguenti versioni del sistema operativo guest:

- Solaris 10 (a partire dall'aggiornamento 4, 08/07) e da Solaris 11.1 a Solaris 11.4

Per Charon-SSP/4v (+), tenete presente quanto segue:

- Per lo SPARC T4 emulato, le versioni di Solaris 10 supportate sono: Oracle Solaris 10 1/13, Oracle Solaris 10 8/11 e Solaris 10 9/10 o Solaris 10 10/09 con il set di patch Oracle Solaris 10 8/11.
- Il modello SPARC T4 emulato è un prerequisito per l'esecuzione di Solaris 11.4 nell'emulatore.
- Le zone del kernel di Solaris non sono supportate.



## Prerequisiti dell'istanza cloud Charon-SSP

Selezionando un tipo o una forma di istanza, si seleziona l'hardware virtuale che verrà utilizzato per l'istanza host Charon-SSP nel cloud. Pertanto, la selezione del tipo o della forma dell'istanza determina le caratteristiche hardware dell'host virtuale Charon-SSP (ad esempio, quanti core CPU e quanta memoria avrà il sistema host virtuale Charon).

### Note

Se utilizzi un'immagine del marketplace Charon-SSP per avviare l'istanza, tutti i requisiti del sistema operativo host Linux sono soddisfatti.

I requisiti hardware minimi sono descritti di seguito.

Punti importanti relativi alle linee guida per il dimensionamento:

- Le linee guida sul dimensionamento riportate di seguito, in particolare per quanto riguarda il numero di core della CPU host e la memoria host, mostrano i requisiti minimi. Ogni situazione di implementazione deve essere esaminata e il dimensionamento effettivo dell'host deve essere adattato, se necessario. Ad esempio, il numero di core CPU disponibili per l'I/O deve essere aumentato se le applicazioni guest producono un carico di I/O elevato. Inoltre, un sistema con molte CPU emulate è in genere in grado di creare un carico di I/O più elevato e pertanto potrebbe essere necessario aumentare il numero di core CPU disponibili per l'I/O. In un ambiente hyper-threading, per ottenere prestazioni ottimali, il numero di core CPU (ovvero CPU reali/fisiche) deve essere sufficiente a soddisfare i requisiti di CPU degli emulatori attivi, evitando così che i thread con carichi di lavoro elevati condividano un core fisico della CPU.
- L'allocazione dei core della CPU per le CPU emulate e i core CPU per l'elaborazione I/O è determinata dalla configurazione. Vedi Configurazione della CPU nella Guida generale dell'utente di Charon-SSP per ulteriori informazioni su questa e sull'allocazione predefinita dei core della CPU per l'elaborazione I/O.

### Informazioni generali importanti

- Per facilitare un trasferimento rapido dei dati dell'emulatore da un'istanza cloud a un'altra, si consiglia vivamente di archiviare tutti i dati pertinenti dell'emulatore su un volume disco

separato che può essere facilmente rimosso dalla vecchia istanza e collegato a una nuova istanza.

- Assicurati di dimensionare correttamente l'istanza sin dall'inizio (verifica i requisiti minimi di seguito). La licenza Charon-SSP per Charon-SSP AL viene creata al primo avvio dell'istanza. Passare successivamente a un'altra dimensione/tipo di istanza e quindi modificare il numero di core della CPU invaliderà la licenza e quindi impedirà l'avvio delle istanze Charon (è necessaria una nuova istanza). Se intendi utilizzare l'istanza Charon-SSP AL in modalità AutoVe, assicurati di includere le informazioni sul server AutoVe prima del primo avvio, altrimenti verranno utilizzati i server di licenza pubblici. La licenza per Charon-SSP VE viene creata sulla base dell'impronta digitale acquisita sul server delle licenze. Se il server di licenza viene eseguito direttamente sull'host dell'emulatore e l'host dell'emulatore richiede successivamente, ad esempio, una modifica del numero di core della CPU, la licenza verrà invalidata (è necessaria una nuova licenza ed eventualmente una nuova istanza).

## Prerequisiti dell'istanza

Requisiti generali della CPU: Charon-SSP supporta i moderni processori con architettura x86-64 basati su istanze Amazon EC2.

Requisiti minimi per Charon-SSP:

- Numero minimo di core CPU del sistema host:
  - Almeno un core di CPU per il sistema operativo host, più:
  - Per ogni sistema SPARC emulato:
    - Un core di CPU per ogni CPU emulata dell'istanza, più:
    - Almeno un core CPU aggiuntivo per l'elaborazione I/O (almeno due, se si utilizza l'ottimizzazione JIT del server). Vedi la sezione Configurazione della CPU menzionata sopra per le opzioni di configurazione. Per impostazione predefinita, Charon assegnerà 1/3 (min. 1; arrotondato per difetto) del numero di CPU visibili all'host Charon all'elaborazione I/O.
- Requisiti minimi di memoria:
  - 4 GB o più di RAM per il sistema operativo host Linux. I requisiti effettivi potrebbero essere più elevati e dipenderanno dai requisiti dei servizi non emulatori in esecuzione sull'host Linux. La raccomandazione precedente di almeno 2 GB di RAM per l'host Linux sarà ancora valida per

molti sistemi, ma i crescenti requisiti del sistema operativo e delle applicazioni Linux hanno portato alla raccomandazione aggiornata per le nuove installazioni. Inoltre:

- Per ogni sistema SPARC emulato:
  - La memoria configurata dell'istanza emulata, oltre a:
  - 2 GB di RAM (6 GB di RAM se si utilizza il server JIT) per consentire l'ottimizzazione DIT, i requisiti dell'emulatore, i buffer di runtime, l'SMP e l'emulazione grafica.
- Se l'hyper-threading è abilitato sulle moderne CPU x86-64, è possibile eseguire due thread su un core fisico della CPU fornendo due CPU logiche al sistema operativo host. Se possibile, disabilita l'hyper-threading sull'host Charon-SSP. Tuttavia, spesso ciò non è possibile in ambienti VMware e cloud, oppure non è chiaro se venga utilizzato o meno l'hyper-threading. L'opzione hyper-threading Charon-SSP consente a Charon-SSP di adattarsi a tali ambienti. Consulta la sezione Configurazione della CPU nella Guida per l'utente generale di Charon-SSP menzionata sopra per informazioni dettagliate sulla configurazione. Nota: per ottenere prestazioni ottimali, i thread Charon-SSP non devono condividere un core fisico della CPU: sul sistema host dovrebbero essere disponibili abbastanza core fisici per soddisfare i requisiti degli emulatori configurati.
- Una o più interfacce di rete, a seconda delle esigenze del cliente.
- Charon-SSP/4U+ e Charon-SSP/4v+ devono funzionare su hardware fisico che supporti Intel VT-X/EPT o AMD-V/NPT (istanze baremetal) e pertanto non possono essere eseguiti in tutti gli ambienti cloud. Consulta la documentazione del tuo provider di servizi cloud per verificare la disponibilità di tale hardware. Inoltre, tieni presente i seguenti punti:
  - Charon-SSP/4U+ e Charon-SSP/4V+ sono disponibili solo quando si utilizza un kernel Linux supportato da Stromasys.
  - Se avete bisogno di questo tipo di hardware SPARC emulato, contattate Stromasys o il vostro Stromasys VAR per discutere dettagliatamente le vostre esigenze.

## Creazione e configurazione di un'istanza AWS cloud per Charon (Nuova GUI)

Questa sezione riflette la primavera del 2022 AWS Management Console . Se utilizzi ancora la console precedente, consulta l'appendice della guida introduttiva di Charon-SSP AWS .

### Prerequisiti generali

Questa descrizione mostra la configurazione di base di un'istanza Linux in AWS. Non elenca prerequisiti specifici. Tuttavia, a seconda del caso d'uso, considera i seguenti prerequisiti:

- Account Amazon e Marketplace AWS abbonamenti
  - Per configurare un'istanza Linux in AWS, è necessario un AWS account con accesso da amministratore.
  - Identifica la AWS regione in cui intendi avviare l'istanza. Assicurati che AWS i servizi che intendi utilizzare siano disponibili in quella regione. Vedi [AWS Servizi per regione](#).
  - Identifica il VPC e la sottorete in cui intendi avviare l'istanza.
  - Se la tua istanza richiede l'accesso a Internet, assicurati che la tabella di routing associata al tuo VPC disponga di un gateway Internet. Se la tua istanza richiede l'accesso VPN alla rete locale, assicurati che sia disponibile un gateway VPN. L'esatta configurazione del VPC e delle relative sottoreti dipenderà dalla progettazione della rete e dai requisiti dell'applicazione.
  - Per abbonarti a un Marketplace AWS servizio specifico, scegli AWS Marketplace Subscriptions nel AWS Management Console campo, quindi scegli Gestisci abbonamenti.
  - Cerca il servizio che intendi utilizzare e abbonati. Dopo un abbonamento andato a buon fine, troverai l'abbonamento nella sezione Gestisci gli abbonamenti. Da lì puoi avviare direttamente una nuova istanza.
- I prerequisiti hardware e software dell'istanza saranno diversi a seconda dell'uso pianificato dell'istanza:
  - Opzione 1: l'istanza deve essere utilizzata come sistema host dell'emulatore Charon:
    - Fate riferimento alle sezioni sui prerequisiti hardware e software della Guida per l'utente e/o della guida introduttiva del prodotto Charon per determinare gli esatti prerequisiti hardware e software che devono essere soddisfatti dall'istanza Linux. L'immagine utilizzata per avviare l'istanza e il tipo di istanza scelto determinano il software e l'hardware dell'istanza cloud.
    - È necessaria una licenza del prodotto Charon per eseguire sistemi legacy emulati. Fate riferimento alle informazioni sulla licenza contenute nella documentazione del prodotto Charon o contattate il vostro rappresentante Stromasys o Stromasys VAR per ulteriori informazioni.
  - Opzione 2: l'istanza deve essere utilizzata come server di licenza VE dedicato:
    - Consulta la VE License Server Guide per i prerequisiti dettagliati.
- Alcuni sistemi operativi legacy che possono essere eseguiti nei sistemi emulati forniti dai prodotti di emulazione Charon richiedono una licenza del fornitore originale del sistema operativo. L'utente è responsabile di eventuali obblighi di licenza relativi al sistema operativo legacy e deve fornire le licenze appropriate.

## Utilizzo di AWS Management Console per avviare una nuova istanza

Per creare una nuova istanza

1. [Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.](https://console.aws.amazon.com/ec2/)
2. Scegliere Launch Instance (Avvia istanza).
3. Inserisci un nome per l'istanza.
4. Selezionare un'AMI. Un'AMI è un'immagine preconfezionata utilizzata per avviare istanze cloud. Include il sistema operativo e il software applicativo applicabile. La scelta dell'AMI dipende dal modo in cui intendi utilizzare l'istanza:
  - Se l'istanza deve essere utilizzata come sistema host dell'emulatore Charon, sono possibili diverse scelte AMI:
    - Installazione del sistema host Charon da un'immagine Charon Marketplace preconfezionata: contengono il sistema operativo sottostante e il software Charon preinstallato.
    - Verificate con il vostro rappresentante Stromasys quali opzioni sono attualmente disponibili nel marketplace del vostro provider di servizi cloud.
    - A seconda del provider di servizi cloud e dei piani di rilascio del prodotto Stromasys, possono esserci due varianti:
      - Licenze automatiche (AL) da utilizzare con un server di licenze pubblico gestito da Stromasys o con un server di licenze AutoVe privato gestito dal cliente
      - Ambiente virtuale (VE) da utilizzare con un server di licenze VE privato gestito dal cliente
  - Installazione del sistema host Charon utilizzando un'installazione convenzionale dell'emulatore Charon con i pacchetti RPM di installazione dell'emulatore Charon per Linux:
    - Scegli un'AMI Linux di una distribuzione supportata dal prodotto e dalla versione Charon selezionati. Consultate la guida per l'utente del prodotto sul sito di documentazione di Stromasys.
- Se l'istanza deve essere utilizzata come server di licenza VE dedicato, consulta la VE License Server Guide nella documentazione sulle licenze per i requisiti dell'istanza Linux.

Dopo aver deciso quale AMI è richiesta, seleziona un AMI del prodotto Linux o Charon corrispondente. Se non vedi l'AMI che ti serve, scegli Sfoglia altre AMI. Scegli l'AMI Linux che corrisponde al modo in cui intendi utilizzare l'istanza. Può essere uno dei seguenti:

- Un'immagine preconfezionata del marketplace Charon VE. Il nome dell'AMI includerà la stringa «ve».
  - Un'immagine del marketplace Charon AL preconfezionata per Automatic Licensing o AutoVe.
  - Una versione Linux supportata per l'installazione di un prodotto RPM.
  - Una versione Linux supportata per il server di licenza VE.
5. Selezione di un tipo di istanza. Amazon EC2 offre tipi di istanze con diverse combinazioni di CPU, memoria, storage e capacità di rete. Seleziona un tipo di istanza che soddisfi i requisiti del prodotto Charon che desideri utilizzare. Alcune immagini del marketplace hanno una selezione limitata di tipi di istanze.
6. Seleziona una key pair esistente o creane e salvane una nuova. Se selezioni una coppia di chiavi esistente, assicurati di avere la chiave privata corrispondente. Altrimenti, non sarai in grado di connetterti alla tua istanza.

#### Note

Se il tuo sistema di gestione lo supporta, per RHEL 9.x, Rocky Linux 9.x e Oracle Linux 9.x, usa la chiave SSH di tipo ECDSA o ED25519. Questi tipi consentono di connettersi a questi sistemi Linux host Charon utilizzando un tunnel SSH senza dover modificare le impostazioni di crittografia predefinite sull'host Charon con impostazioni meno sicure. Ad esempio, questo è importante per Charon-SSP Manager. Vedi [Utilizzo di policy crittografiche a livello di sistema](#) nella documentazione di Red Hat.

7. Nella sezione Impostazioni di rete, scegliete Modifica. Scegli le impostazioni che corrispondono al tuo ambiente.
- Specificare un VPC.
  - Specificate una sottorete esistente o createne una nuova.
  - Abilita o disabilita l'assegnazione automatica di un indirizzo IP pubblico all'interfaccia principale. L'assegnazione automatica è possibile solo se l'istanza ha un'unica interfaccia di rete.
  - Assegna un gruppo di sicurezza personalizzato esistente o nuovo. Il gruppo di sicurezza deve consentire almeno a SSH di accedere all'istanza. Inoltre, devono essere consentite tutte le porte richieste dalle applicazioni che si prevede di eseguire sull'istanza. È possibile modificare il gruppo di sicurezza in qualsiasi momento dopo aver creato l'istanza.

- Nella sezione Archiviazione, per il volume principale (il disco di sistema), scegli una dimensione appropriata per il tuo ambiente. La dimensione minima consigliata del disco di sistema per il sistema Linux è 30 GiB. Per fornire spazio per contenitori di dischi virtuali e altri requisiti di archiviazione, puoi aggiungere altro spazio di archiviazione ora o dopo aver avviato l'istanza. Tuttavia, le dimensioni del disco di sistema devono coprire i requisiti di sistema Linux, incluse tutte le applicazioni e le utilità che intendi installare.

 Note

Si consiglia di creare volumi di archiviazione separati per i dati delle applicazioni Charon (ad esempio, immagini del disco). Se necessario, è possibile migrare successivamente tali volumi su un'altra istanza.

- Espandi la sezione Dettagli avanzati, scorri verso il basso e seleziona Specificare le opzioni della CPU. Tre esempi che hanno maggiori probabilità di essere utili in un ambiente di emulazione Charon sono mostrati nell'immagine seguente.



Specify CPU options

Core count

2

Threads per core

2

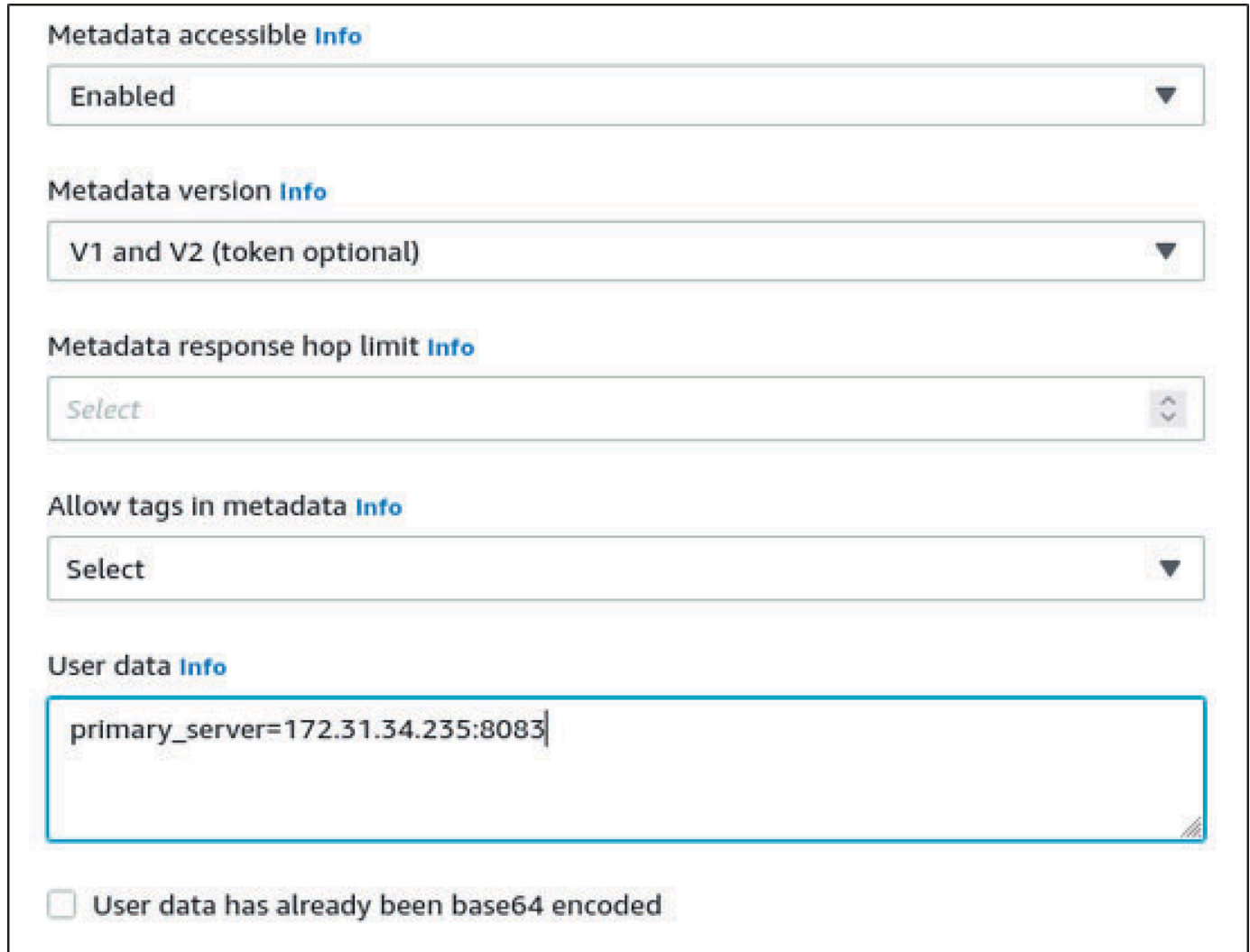
Number of vCPUs

4

- Per un sistema server di licenza VE con una versione precedente alla 1.1.23, è necessario assegnare il ruolo IAM richiesto all'istanza. Deve essere un ruolo che consenta l'azione. ListUsers Per assegnare un ruolo, nella sezione estesa dei dettagli avanzati seleziona un ruolo nel profilo dell'istanza IAM o scegli Crea un nuovo profilo IAM. Per ulteriori informazioni consulta [Ruoli IAM per Amazon EC2](#).

11. Se la tua istanza è basata su un' Marketplace AWS immagine Charon AL e prevedi di utilizzare i server con licenza pubblica gestiti da StromaSys, devi aggiungere le informazioni corrispondenti alla configurazione dell'istanza prima di avviare l'istanza.

Immettete le informazioni per il server di licenza AutoVe come mostrato nell'immagine seguente.



The image shows a configuration interface with the following elements:

- Metadata accessible** Info: A dropdown menu set to "Enabled".
- Metadata version** Info: A dropdown menu set to "V1 and V2 (token optional)".
- Metadata response hop limit** Info: A dropdown menu set to "Select".
- Allow tags in metadata** Info: A dropdown menu set to "Select".
- User data** Info: A text input field containing the text `primary_server=172.31.34.235:8083`.
- User data has already been base64 encoded**

Le seguenti sono opzioni di configurazione dei dati utente valide:

- **primary\_server**=<ip-address>[:<port>]
- **backup\_server**=<ip-address>[:<port>]

Dove

- <ip-address> indica l'indirizzo IP del server primario e del server di backup, a seconda dei casi.



- <port>indica una porta TCP non predefinita utilizzata per comunicare con il server di licenza (impostazione predefinita: TCP/8083).

### Note

Almeno un server di licenza deve essere configurato all'avvio iniziale per abilitare la modalità AutoVe. In caso contrario, l'istanza si collegherà a uno dei server di licenza pubblici gestiti da Stromasys.

12. Nella sezione Riepilogo, scegli Launch instance. Dopo un po', vedrai il seguente messaggio di successo:

The screenshot shows the AWS Management Console interface. At the top, a dark navigation bar contains a hamburger menu icon, the text 'EC2 > Instances > Launch an instance', and a search icon. Below this, a green notification banner with a checkmark icon displays the text: 'Success Successfully initiated launch of instance (i-01304600000000000)'. Underneath the notification is a 'Launch log' section with a right-pointing arrow. The main content area is titled 'Next Steps' and features a search bar with the placeholder text: 'What would you like to do next with this instance, for example "create alarm" or "create backup"'. Below the search bar are two white cards with rounded corners. The left card is titled 'Create billing and free tier usage alerts' and contains the text: 'To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.' It has a blue button labeled 'Create billing alerts' with an external link icon. The right card is titled 'Connect to your instance' and contains the text: 'Once your instance is running, log into it from your local computer.' It has a blue button labeled 'Connect to instance' with an external link icon and a blue link labeled 'Learn more' with an external link icon.

13. Nell'angolo in basso a destra dello schermo, scegli Visualizza tutte le istanze.
14. Per visualizzare i dettagli dell'istanza, seleziona la casella di controllo a sinistra della riga che rappresenta l'istanza nella tabella Istanze. I dettagli dell'istanza verranno visualizzati nella metà inferiore dello schermo. Per informazioni su come connetterti alla tua istanza, consulta [Connect](#) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

# AWS Modernizzazione del mainframe: ripiattaforma con NTT DATA

AWS La modernizzazione del mainframe offre una varietà di Amazon Machine Images (AMI). Queste AMI facilitano il provisioning rapido delle istanze Amazon EC2, creando un ambiente personalizzato per il rehosting e la ripiattaforma delle applicazioni mainframe utilizzando NTT Data. AWS Questa guida fornisce i passaggi necessari per accedere e utilizzare queste AMI.

## Prerequisiti

- Assicurati di avere accesso da amministratore a un AWS account in cui puoi creare istanze Amazon EC2.
- Verifica che il servizio di modernizzazione del AWS mainframe sia disponibile nella regione in cui intendi creare le istanze Amazon EC2. Consulta [l'elenco dei servizi AWS disponibili per regione](#).
- Identifica l'Amazon VPC in cui desideri creare le istanze Amazon EC2.

## Iscriviti all'Amazon Machine Image

Quando ti abboni a un prodotto AWS Marketplace, puoi avviare un'istanza dall'AMI del prodotto.

1. Accedi AWS Management Console e apri la Marketplace AWS console all'[indirizzo https://console.aws.amazon.com/marketplace](https://console.aws.amazon.com/marketplace).
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).
3. Copia e incolla il seguente link nella barra degli indirizzi del browser: <https://aws.amazon.com/marketplace/pp/prodview-eg227ymldsrx2>
4. Scegliere Continue to Subscribe (Continua la sottoscrizione).
5. Se i termini e le condizioni sono accettabili, scegli Accetta termini. L'elaborazione dell'abbonamento potrebbe richiedere alcuni minuti.
6. Attendi che venga visualizzato un messaggio di ringraziamento. Questo messaggio conferma che l'iscrizione al prodotto è avvenuta con successo.
7. Nel riquadro di navigazione a sinistra, scegli Gestisci abbonamenti. Questa visualizzazione mostra tutti i tuoi abbonamenti.

# Avvia la ripiattaforma AWS Mainframe Modernization con l'istanza NTT DATA

1. [Apri la console all'indirizzo <https://console.aws.amazon.com/marketplace> Marketplace AWS .](https://console.aws.amazon.com/marketplace)
2. Nel riquadro di navigazione a sinistra, scegli Gestisci abbonamenti.
3. Trova l'AMI che desideri avviare e scegli Launch new instance.
4. In Regione, seleziona la regione consentita.
5. Scegli Continua a lanciare tramite EC2. Questa azione ti porta alla console Amazon EC2.
6. Inserisci un nome per il server.
7. Seleziona un tipo di istanza che soddisfi i requisiti di prestazioni e costi del progetto. Il punto di partenza consigliato per la dimensione dell'istanza è c5.2xLarge.
8. Scegli una key pair esistente o creane e salvane una nuova. Per informazioni sulle coppie di chiavi, consulta le coppie di [chiavi Amazon EC2 e le istanze Linux](#) nella Amazon EC2 User Guide for Linux Instances.
9. Modifica le impostazioni di rete e scegli il VPC nell'elenco consentito e la sottorete appropriata.
10. Scegli un gruppo di sicurezza esistente o creane uno nuovo. Se si tratta di un'istanza Amazon EC2 di Enterprise Server, è normale consentire il traffico TCP verso le porte 86 e 10086 per amministrare la configurazione Micro Focus.
11. Configura lo storage per l'istanza Amazon EC2.
12. Controlla il riepilogo e scegli Launch instance. Affinché il lancio abbia successo, il tipo di istanza deve essere valido. Se l'avvio fallisce, scegli Modifica configurazione dell'istanza e scegli un tipo di istanza diverso.
13. Dopo aver visualizzato il messaggio di successo, scegli Connect to instance.
14. Apri la console Amazon EC2 all'indirizzo <https://console.aws.amazon.com/ec2/>.
15. Nel riquadro di navigazione a sinistra, nel menu Istanze, scegli Istanze.
16. Nel riquadro principale, controlla lo stato della tua istanza.

## Guida introduttiva a NTT Data

Dopo aver effettuato il provisioning dell'istanza Amazon EC2, accedi tramite SSH con il nome utente `ec2-user`. La schermata sarà simile all'immagine seguente.



Dopo aver convalidato con successo l'istanza Amazon EC2, inizia a AWS utilizzare Mainframe Modernization Replatform with NTT DATA seguendo la documentazione di NTT Data.

# Applicazioni nella modernizzazione del AWS mainframe

Se non conosci la modernizzazione del AWS mainframe, consulta i seguenti argomenti per iniziare:

- [Cos'è la modernizzazione del AWS mainframe?](#)
- [Configurazione della modernizzazione del AWS mainframe](#)
- [Tutorial: Managed Runtime per AWS Blu Age](#)
- [Tutorial: runtime gestito per Micro Focus](#)

Un'applicazione in AWS Mainframe Modernization contiene un carico di lavoro mainframe migrato. L'applicazione è analoga a un carico di lavoro sul mainframe ed è associata a un ambiente di runtime. È possibile aggiungere file batch e set di dati alle applicazioni e monitorare le applicazioni durante l'esecuzione. Crei applicazioni di modernizzazione AWS del mainframe per ogni carico di lavoro da migrare. Quando si crea un'applicazione di modernizzazione AWS del mainframe, si specifica il motore su cui viene eseguita l'applicazione al momento della creazione. Scegliete AWS Blu Age se utilizzate il pattern di refactoring automatico e scegliete Micro Focus se utilizzate il pattern di replatforming.

## Argomenti

- [Create un'applicazione di modernizzazione del mainframe AWS](#)
- [Implementa un'applicazione di AWS modernizzazione del mainframe](#)
- [Aggiornate un'applicazione di modernizzazione del AWS mainframe](#)
- [Eliminare un'applicazione di modernizzazione AWS del mainframe da un ambiente](#)
- [Eliminare un'applicazione di modernizzazione AWS del mainframe](#)
- [Invia lavori in batch per le applicazioni di modernizzazione AWS del mainframe](#)
- [Importa set di dati per le applicazioni di modernizzazione AWS del mainframe](#)
- [Gestisci le transazioni per le applicazioni di modernizzazione del AWS mainframe](#)
- [Crea AWS risorse per un'applicazione migrata](#)
- [Configurare l'applicazione gestita](#)
- [AWS Riferimento alla definizione delle applicazioni di modernizzazione del mainframe](#)
- [AWSRiferimento alla definizione del set di dati per la modernizzazione del mainframe](#)

# Create un'applicazione di modernizzazione del mainframe AWS

Utilizzate la console di modernizzazione del AWS mainframe per creare un'AWS applicazione di modernizzazione del mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

## Creazione di un'applicazione

Per creare un'applicazione

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui desiderate creare l'applicazione.
3. Nella pagina Applications (Applicazioni), scegliere Create application (Crea applicazione).
4. Nella pagina Specificare le informazioni di base, nella sezione Nome e descrizione, immettete un nome per l'applicazione.
5. (Facoltativo) Nel campo Descrizione dell'applicazione, immettere una descrizione per l'applicazione. Questa descrizione può aiutare te e gli altri utenti a identificare lo scopo dell'applicazione.
6. Nella sezione Tipo di motore, scegliete Blu Age per il refactoring automatizzato o Micro Focus per il replatforming.
7. Nella sezione Chiave KMS, scegli Personalizza le impostazioni di crittografia se desideri utilizzare una chiave gestita dal cliente. AWS KMS Per ulteriori informazioni, consulta [Crittografia dei dati inattiva per il servizio di AWS modernizzazione del mainframe](#).

### Note

Per impostazione predefinita, AWS Mainframe Modernization crittografa i tuoi dati con una AWS KMS chiave che AWS Mainframe Modernization possiede e gestisce per te. Tuttavia, puoi scegliere di utilizzare una chiave gestita dal cliente. AWS KMS

8. (Facoltativo) Scegli una AWS KMS chiave per nome o Amazon Resource Name (ARN) oppure scegli Crea una AWS KMS chiave per accedere alla AWS KMS console e creare una nuova AWS KMS chiave.

9. (Facoltativo) Nella sezione Tag, scegli Aggiungi nuovo tag per aggiungere uno o più tag di applicazione alla tua applicazione. Un tag di applicazione è un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse).
10. Seleziona Avanti.
11. Nella sezione Risorse e configurazioni, utilizzate l'editor in linea per inserire la definizione dell'applicazione. In alternativa, scegli Usa un file JSON di definizione dell'applicazione in un bucket Amazon S3 e fornisci la posizione della definizione dell'applicazione che desideri utilizzare. Per ulteriori informazioni, consulta [AWS Esempio di definizione dell'applicazione Blu Age](#) o [Definizione dell'applicazione Micro Focus](#).
12. Seleziona Avanti.
13. Nella pagina Rivedi e crea, esamina le informazioni che hai inserito, quindi scegli Crea applicazione.

## Implementa un'applicazione di AWS modernizzazione del mainframe

Utilizza la console di modernizzazione del AWS mainframe per distribuire un'AWS applicazione di modernizzazione del mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

### Implementa un'applicazione

Per eseguire un'applicazione di modernizzazione del AWS mainframe, è necessario prima distribuirla in un ambiente di runtime. Un'applicazione può avere più di una versione. Ogni versione di un'applicazione ha una propria definizione di applicazione. Per distribuire un'applicazione, è necessario specificare la versione che si desidera distribuire.

È possibile distribuire solo una versione di una determinata applicazione alla volta. Se distribuisce una versione di un'applicazione e poi decidi di distribuire una versione diversa, devi prima arrestare l'applicazione se è in esecuzione.

Per distribuire un'applicazione

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).



2. Nel Regione AWS selettore, scegliete la regione in cui desiderate creare l'applicazione.
3. Nella pagina Applicazioni, scegli l'applicazione che desideri distribuire.
4. Scegli Distribuisci applicazione.
5. Nella sezione Versioni disponibili, scegli la versione che desideri distribuire.
6. Nella sezione Ambienti, scegli un ambiente di runtime in cui desideri che l'applicazione venga eseguita.
7. Seleziona Deploy (Implementa).

Per distribuire una versione diversa di un'applicazione distribuita

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui desiderate creare l'applicazione.
3. Nella pagina Applicazioni, scegli l'applicazione che desideri distribuire.
4. Dal menu Azioni, scegli Arresta applicazione.
5. Dopo l'arresto dell'applicazione, scegli Distribuisci applicazione.
6. Nella sezione Versioni disponibili, scegli la versione che desideri distribuire. Nella sezione Ambienti, l'ambiente in cui l'applicazione è già distribuita è preselezionato.
7. Seleziona Deploy (Implementa).

## Aggiornate un'applicazione di modernizzazione del AWS mainframe

Utilizzate la console di modernizzazione del AWS mainframe per aggiornare un'AWSApplicazione di modernizzazione del mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

### Aggiornamento di un'applicazione

Un'applicazione di modernizzazione AWS del mainframe può avere più versioni, ognuna con la propria definizione di applicazione. Per aggiornare un'applicazione, fornite una nuova definizione di applicazione. In questo modo viene creata una nuova versione dell'applicazione.

## Per aggiornare un'applicazione

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stata creata l'applicazione che desiderate aggiornare.
3. Nella pagina Applicazioni, scegli l'applicazione che desideri aggiornare.
4. Nella pagina dei dettagli dell'applicazione, nella sezione Definizione corrente, scegli Modifica per aggiornare la definizione corrente dell'applicazione.
5. Nella pagina Aggiorna applicazione, utilizzate l'editor in linea per aggiornare la definizione corrente dell'applicazione.

In alternativa, scegli Usa un file JSON di definizione dell'applicazione in un bucket Amazon S3 e fornisci la posizione della definizione dell'applicazione che desideri utilizzare. Per ulteriori informazioni, consulta [AWS Esempio di definizione dell'applicazione Blu Age](#) o [Definizione dell'applicazione Micro Focus](#).

6. Quando hai finito di aggiornare la definizione dell'applicazione, scegli Aggiorna.

### Note

Dopo aver aggiornato l'applicazione, è necessario distribuirla nuovamente. Per ulteriori informazioni, consulta [Implementa un'applicazione di AWS modernizzazione del mainframe](#).

## Eliminare un'applicazione di modernizzazione AWS del mainframe da un ambiente

È possibile eliminare un'applicazione di modernizzazione AWS del mainframe da un ambiente utilizzando la console di modernizzazione del AWS mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

## Eliminare un'applicazione da un ambiente

Se devi eliminare un'applicazione di modernizzazione del AWS mainframe che è in esecuzione, assicurati di interromperla prima. È possibile visualizzare lo stato dell'applicazione nella pagina Applicazioni.

Per eliminare un'applicazione da un ambiente

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stata creata l'applicazione che desiderate eliminare dall'ambiente.
3. Nella pagina Applicazioni, scegli l'applicazione che desideri eliminare dall'ambiente, quindi scegli Azioni.
4. (Facoltativo) Se lo stato dell'applicazione è Running, scegli Interrompi applicazione.
5. Scegli Elimina dall'ambiente.

Il processo di eliminazione inizia immediatamente.

## Eliminare un'applicazione di modernizzazione AWS del mainframe

Utilizzate la console di modernizzazione del AWS mainframe per eliminare un'AWS applicazione di modernizzazione del mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

### Eliminazione di un'applicazione

Se devi eliminare un'applicazione di modernizzazione del AWS mainframe che è in esecuzione, assicurati di interromperla prima. È possibile visualizzare lo stato dell'applicazione nella pagina Applicazioni.

Eliminazione di un'applicazione

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).

2. Nel Regione AWS selettore, scegliete la regione in cui è stata creata l'applicazione che desiderate eliminare.
3. Nella pagina Applicazioni, scegli l'applicazione che desideri eliminare, quindi scegli Azioni.
4. (Facoltativo) Se lo stato dell'applicazione è Running, scegli Interrompi applicazione.
5. Scegli Elimina applicazione.
6. Nella finestra Elimina applicazione, immettete delete per confermare che desiderate eliminare l'applicazione, quindi scegliete Elimina.

## Invia lavori in batch per le applicazioni di modernizzazione AWS del mainframe

In AWS Mainframe Modernization puoi inviare lavori in batch per le tue applicazioni. È possibile inviare o annullare i lavori in batch e rivedere i dettagli sulle esecuzioni dei lavori in batch. Ogni volta che si invia un processo batch, AWS Mainframe Modernization crea un'esecuzione del lavoro batch separata. È possibile monitorare l'esecuzione di questo lavoro. È possibile cercare i processi batch per nome e fornire file JCL o di script ai lavori batch.

### Important

Se si annulla un processo batch, ciò non elimina il lavoro. Annulla una particolare esecuzione del processo batch. I record del processo batch rimangono disponibili per la visualizzazione nei dettagli relativi all'esecuzione del processo batch.

Se il processo batch richiede l'accesso a uno o più set di dati, utilizzate la console di modernizzazione del AWS mainframe o AWS Command Line Interface (AWS CLI) per importare i set di dati. Per ulteriori informazioni, consulta [Importa set di dati per le applicazioni di modernizzazione AWS del mainframe](#).

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali [Configurazione della modernizzazione del AWS mainframe](#) e successivi. [Create un'applicazione di modernizzazione del mainframe AWS](#)

## Invia un lavoro in batch

Per inviare un lavoro in batch

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stata creata l'applicazione per la quale desiderate inviare un processo batch.
3. Nella pagina Applicazioni, scegli l'applicazione per la quale desideri inviare un processo in batch.

### Note

Prima di poter inviare un processo batch a un'applicazione, è necessario distribuire correttamente l'applicazione.

4. Nella pagina dei dettagli dell'applicazione, scegli Lavori in Batch.
5. Seleziona Submit job (Invia processo).
6. Nella sezione Seleziona uno script, scegli uno script. Puoi cercare lo script che desideri per nome.
7. Seleziona Submit job (Invia processo).

## Importa set di dati per le applicazioni di modernizzazione AWS del mainframe

Con AWS Mainframe Modernization puoi importare set di dati da utilizzare con le tue applicazioni. Puoi specificare i set di dati in un file JSON archiviato in un bucket Amazon S3 oppure puoi specificare i valori di configurazione dei set di dati separatamente. Dopo aver importato i set di dati, puoi rivedere i dettagli dell'attività di importazione per confermare che i set di dati che desideri siano stati importati. Tutti i set di dati catalogati per un'applicazione sono elencati insieme nella console.

Utilizzate la console di modernizzazione del AWS mainframe per importare set di dati per un'applicazione di modernizzazione del AWS mainframe.

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali e successivi.

[Configurazione della modernizzazione del AWS mainframe](#) [Create un'applicazione di modernizzazione del mainframe AWS](#)

## Importa un set di dati

Per importare un set di dati

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stata creata l'applicazione per cui desiderate importare i set di dati.
3. Nella pagina Applicazioni, scegli l'applicazione per cui desideri importare i set di dati.
4. Nella pagina dei dettagli dell'applicazione, scegli Set di dati.
5. Seleziona Importa.
6. Esegui una di queste operazioni:
  - Scegli Usa il file JSON di configurazione del set di dati in un bucket Amazon S3 e fornisci la posizione della configurazione del set di dati.
  - Scegli Specificare separatamente i valori di configurazione del set di dati con la configurazione guidata. Fate riferimento [the section called “Riferimento alla definizione del set di dati”](#) per i dettagli specifici della definizione.

Inserisci il nome, l'organizzazione del set di dati (VSAM, GDG, PO, PS), la posizione e la posizione esterna di Amazon S3 e le impostazioni dei parametri per ogni valore di configurazione del set di dati. Nella configurazione guidata puoi anche scegliere Generate JSON per esaminare la configurazione JSON in base al tuo input.

7. Seleziona Invia.

## Gestisci le transazioni per le applicazioni di modernizzazione del AWS mainframe

Con AWS Mainframe Modernization è possibile eseguire un'applicazione, su richiesta, contemporaneamente a molti altri utenti che inviano richieste per eseguire la stessa applicazione utilizzando gli stessi file e programmi. Una singola transazione consiste in uno o più programmi applicativi che eseguono l'elaborazione necessaria.

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali [Configurazione della modernizzazione del AWS mainframe](#) e successivi [Create un'applicazione di modernizzazione del mainframe AWS](#).

## Gestisci le transazioni per le applicazioni

È possibile visualizzare e modificare le transazioni per le applicazioni.

Per gestire le transazioni per le applicazioni

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
  2. Nel Regione AWS selettore, scegliete Regione AWS dove è stata creata l'applicazione che desiderate eseguire.
  3. Nella pagina Applicazioni, scegli l'applicazione per cui desideri gestire le transazioni.
  4. Nella scheda Transazioni, in Risorse per le transazioni, scegli come desideri che vengano visualizzate le tue risorse dall'elenco a discesa. È possibile visualizzare le risorse in base alle risorse delle transazioni, ai gruppi, agli elenchi o ai SIT.
- Le risorse di transazione consentono di scegliere il tipo di risorsa in base alle definizioni dei file, alle definizioni delle transazioni, alle definizioni dei programmi o alle definizioni delle code di dati transitorie.

### Note

La transazione AWS Mainframe Modernization Managed supporta più tipi di risorse rispetto a questi, ma non è possibile modificarli direttamente qui. Le altre risorse devono essere modificate esternamente ed è necessario ricreare l'applicazione con voci di risorse aggiornate.

- I gruppi sono una raccolta di risorse di transazione. Puoi anche scegliere i gruppi da associare alla tua risorsa di transazione.
- Gli elenchi sono una raccolta ordinata di gruppi. È possibile visualizzare tutte le risorse e i gruppi relativi alle transazioni in una visualizzazione a elenco. L'elenco di avvio determina quali risorse vengono caricate quando il server viene inizializzato.
  - Con il motore di refattore Blu Age, si specificano gli elenchi da includere all'avvio e non c'è limite al numero di elenchi.
  - Con il motore di ripiattaforma Micro Focus, è possibile specificare fino a quattro elenchi in un SIT.

- SIT (System Initialization Table) mostra tutte le configurazioni di transazione disponibili. È possibile trovare i SIT in base alle proprietà (nome, descrizione ed elenchi di avvio). È inoltre possibile scegliere elenchi da associare al SIT prescelto.

#### Note

I SIT sono applicabili solo per il motore di ripiattaforma Micro Focus.

5. Scegliete una risorsa di transazione per visualizzare tutte le informazioni sulla risorsa. È inoltre possibile visualizzare tutti gli attributi associati alla risorsa di transazione e visualizzare o modificare eventuali attributi aggiuntivi.
6. Se desideri modificare qualsiasi informazione relativa alla risorsa di transazione corrente, scegli Modifica.
  - a. Nella pagina Modifica, puoi aggiungere o modificare la descrizione della risorsa.
    1. Per le risorse di transazione visualizzate negli elenchi, puoi anche aggiungere, rimuovere o riordinare gli elenchi come desideri.
    2. Per tipi di risorse specifici (definizioni di file, definizioni di transazioni, definizioni di programmi e definizioni di code di dati transitorie), è possibile modificare le proprietà delle singole risorse. Queste proprietà variano in base al motore e al tipo di risorsa.
  - b. Dopo aver apportato le modifiche desiderate, scegliete Salva modifiche.

Quando la risorsa viene aggiornata correttamente, viene visualizzato un messaggio.

## Crea AWS risorse per un'applicazione migrata

Per eseguire l'applicazione migrata in AWS, è necessario creare alcune AWS risorse insieme ad altre. Servizi AWS Le risorse da creare includono quanto segue:

- Un bucket S3 per contenere il codice dell'applicazione, la configurazione, i file di dati e altri elementi richiesti.
- Un database Amazon RDS o Amazon Aurora per contenere i dati richiesti dall'applicazione.
- Un AWS KMS key, necessario AWS Secrets Manager per creare e archiviare segreti.
- Un segreto di Secrets Manager per contenere le credenziali del database.



**Note**

Ogni applicazione migrata richiede il proprio set di queste risorse. Si tratta di un set minimo. La tua applicazione potrebbe richiedere anche risorse aggiuntive, come i segreti di Amazon Cognito o le code MQ.

## Autorizzazioni richieste

Assicurati di disporre delle seguenti autorizzazioni:

- `s3:CreateBucket`, `s3:PutObject`
- `rds:CreateDBInstance`
- `kms:CreateKey`
- `secretsmanager:CreateSecret`

## Bucket Amazon S3

Sia le applicazioni refactorizzate che quelle riplatformate richiedono un bucket S3 configurato come segue:

```
bucket-name/root-folder-name/application-name
```

### bucket-name

Qualsiasi nome entro i limiti della denominazione di Amazon S3. Ti consigliamo di includere il Regione AWS nome come parte del nome del bucket. Assicurati di creare il bucket nella stessa regione in cui intendi distribuire l'applicazione migrata.

### root-folder-name

Nome richiesto per soddisfare i vincoli della definizione dell'applicazione, creata come parte dell'applicazione Mainframe Modernization. AWS È possibile utilizzare il `root-folder-name` per distinguere tra diverse versioni di un'applicazione, ad esempio V1 e V2.

### application-name

Il nome dell'applicazione migrata, ad esempio, o. PlanetsDemo BankDemo

## Database

Sia le applicazioni rifattorizzate che quelle riplatformate potrebbero richiedere un database. È necessario creare, configurare e gestire il database in base ai requisiti specifici di ciascun motore di runtime. AWS La modernizzazione del mainframe supporta la crittografia in transito su questo database. Se abiliti SSL sul tuo database, assicurati di specificare `sslMode` nel database il segreto insieme ai dettagli di connessione del database. Per ulteriori informazioni, consulta [AWS Secrets Managersegreto](#).

Se utilizzi il pattern di refactoring AWS Blu Age e hai bisogno di un BluSam database, il motore di runtime AWS Blu Age prevede un database PostgreSQL Amazon Aurora, che devi creare, configurare e gestire. Il database è facoltativo. BluSam Crea questo database solo se l'applicazione lo richiede. Per creare il database, segui i passaggi descritti in [Creazione di un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

Se utilizzi il modello di ripiattaforma Micro Focus, puoi creare un database Amazon RDS o Amazon Aurora PostgreSQL. Per creare il database, segui i passaggi in [Creazione di un'istanza DB Amazon RDS](#) nella Guida per l'utente di Amazon RDS o in [Creazione di un cluster Amazon Aurora DB](#) nella Guida per l'utente di Amazon Aurora.

Per entrambi i motori di runtime, devi memorizzare le credenziali del database AWS Secrets Manager utilizzando un per crittografarle. AWS KMS key

## Chiave AWS Key Management Service

È necessario archiviare le credenziali per il database dell'applicazione in modo sicuro. AWS Secrets Manager Per creare un segreto in Secrets Manager, devi creare unAWS KMS key. Per creare una chiave KMS, segui i passaggi descritti in [Creazione di chiavi](#) nella Guida per gli AWS Key Management Service sviluppatori.

Dopo aver creato la chiave, è necessario aggiornare la politica chiave per concedere le autorizzazioni di decrittografia di AWS Mainframe Modernization. Aggiungete le seguenti dichiarazioni politiche:

```
{
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : "kms:Decrypt",
```

```
"Resource" : "*"
}
```

## AWS Secrets Managersegreto

È necessario archiviare le credenziali per il database dell'applicazione in modo sicuro in AWS Secrets Manager. Per creare un segreto, seguite la procedura descritta in [Creare un segreto del database nella Guida](#) per l'AWS Secrets Managerutente.

AWSMainframe Modernization supporta la crittografia in transito su questo database. Se abiliti SSL sul tuo database, assicurati di specificare `sslMode` nel database il segreto insieme ai dettagli di connessione del database. È possibile specificare uno dei seguenti valori per `sslMode`: `verify-fullverify-ca`, `odisable`.

Durante il processo di creazione della chiave, scegli Autorizzazioni per le risorse - opzionale, quindi scegli Modifica autorizzazioni. Nell'editor delle politiche, aggiungi una politica basata sulle risorse, come la seguente, per recuperare il contenuto dei campi crittografati.

```
{
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : "secretsmanager:GetSecretValue",
  "Resource" : "*"
}
```

## Configurare l'applicazione gestita

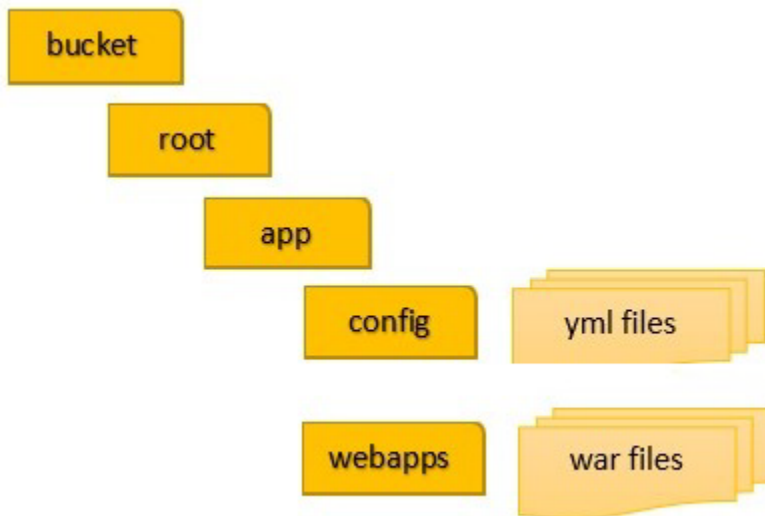
È possibile configurare l'applicazione per includere l'accesso alle utilità legacy. È possibile personalizzare anche proprietà aggiuntive. Per capire cosa è possibile configurare e dove, è utile comprendere la struttura generale di un'applicazione modernizzata AWS Blu Age.

### Argomenti

- [Struttura delle applicazioni gestite da AWS Blu Age](#)
- [Configurazione dell'accesso alle utilità per le applicazioni gestite](#)
- [Aggiungi proprietà di configurazione per il motore AWS Blu Age](#)

## Struttura delle applicazioni gestite da AWS Blu Age

Se utilizzi il pattern di refactoring AWS Blu Age, il motore di runtime AWS Blu Age prevede la seguente struttura all'interno della `application-name` cartella del bucket S3:



### config

Contiene i file YAML per il progetto. Si tratta dei file YAML specifici dell'applicazione, in genere denominati con un nome simile `application-planetsdemo.yaml` e non con il nome fornito e `application-main.yaml` configurato automaticamente da AWS Mainframe Modernization.

### webapp

Contiene i `war` file per l'applicazione. Questi file sono il risultato del processo di modernizzazione.

Un'applicazione può anche avere le seguenti cartelle opzionali:

### jics/sql

Contiene lo `initJics.sql` script che inizializza il database JICS per l'applicazione.

### script

Contiene script di applicazione, che possono anche essere forniti direttamente all'interno dei file.

### war

## sql

Contiene i file SQL dell'applicazione, che possono essere forniti anche direttamente all'interno dei `war` file.

## link

Contiene i file LNK dell'applicazione, che puoi anche fornire direttamente all'interno dei file. `war`

## Gestione del consumo di memoria Java di un'applicazione

Per gestire il consumo di memoria Java per l'applicazione, aggiungete un file di proprietà denominato `tomcat.properties` alla `application-name` cartella. Questo file può avere due proprietà: `xms`, che specifica il consumo minimo di memoria Java e `xmx`, che specifica il consumo massimo di memoria Java. Di seguito è riportato un esempio del contenuto di un file `validotomcat.properties`.

```
xms=512M
xmx=1G
```

I valori specificati per queste due proprietà possono essere in una delle seguenti unità:

- Byte: non specificate un'unità.
- Kilobyte: aggiungi una K al valore.
- Megabyte: aggiungi una M al valore.
- Gigabyte: aggiungi una G al valore.

## Configurazione dell'accesso alle utilità per le applicazioni gestite

Quando esegui il refactoring di un'applicazione mainframe con AWS Blu Age, potrebbe essere necessario fornire supporto per vari programmi di utilità della piattaforma legacy, come IDCAMS, INFUTILB, SORT e così via, se l'applicazione dipende da questi. AWS Blu Age refactoring fornisce questo accesso con un'applicazione web dedicata che viene distribuita insieme alle applicazioni modernizzate. Questa applicazione web richiede un file di configurazione `application-utility-pgm.yml`, che è necessario fornire. Se non fornisci questo file di configurazione, l'applicazione web non può essere distribuita insieme all'applicazione e non sarà disponibile.

## Argomenti

- [Proprietà di configurazione](#)

Questo argomento descrive tutte le possibili proprietà che è possibile specificare nel file di `application-utility-pgm.yml` configurazione, insieme ai relativi valori predefiniti. L'argomento descrive le proprietà obbligatorie e facoltative. L'esempio seguente è un file di configurazione completo. Elenca le proprietà nell'ordine consigliato. È possibile utilizzare questo esempio come punto di partenza per il proprio file di configurazione.

```
# If the datasource support mode is not static-xa, spring JTA transactions
autoconfiguration must be disabled
spring.jta.enabled: false
logging.config: 'classpath:logback-utility.xml'

# Encoding
encoding: cp1047

# Encoding to be used by INFUTILB and DSNUTILB to generate and read SYSPUNCH files
sysPunchEncoding: cp1047

# Utility database access
spring.aws.client.datasources.primary.secret: `arn:aws:secretsmanager:us-
west-2:111122223333:secret:business-FfmXLG`

treatLargeNumberAsInteger: false

# Zoned mode : valid values = EBCDIC_STRICT, EBCDIC_MODIFIED, AS400
zonedMode: EBCDIC_STRICT

jcl.type: mvs

# Unload properties
# For date/time: if use database configuration is enabled, formats are ignored
# For nbi; use hexadecimal syntaxe to specify the byte value
unload:
  sqlCodePointShift: 384
  nbi:
    whenNull: "6F"
    whenNotNull: "00"
  useDatabaseConfiguration: false
  format:
    date: MM/dd/yyyy
    time: HH.mm.ss
```

```
    timestamp: yyyy-MM-dd-HH.mm.ss.SSSSSS
    chunkSize:500
    fetchSize: 500
    varCharIsNull: false
    columnFiller: space

# Load properties
# Batch size for DSNUTILB Load Task
load:
  sqlCodePointShift: 384
  batchSize: 500
  format:
    localDate: dd.MM.yyyy|dd/MM/yyyy|yyyy-MM-dd
    dbDate: yyyy-MM-dd
    localTime: 'HH:mm:ss|HH.mm.ss'
    dbTime: 'HH:mm:ss'

table-mappings:
  TABLE_1_NAME : LEGACY_TABLE_1_NAME
  TABLE_2_NAME : LEGACY_TABLE_2_NAME
```

## Proprietà di configurazione

È possibile specificare le seguenti proprietà nel file di configurazione.

### spring.jta.enabled

Facoltativo. Controlla se il supporto JTA è abilitato. Per le utilità, si consiglia di impostare questo valore su. `false`

```
spring.jta.enabled : false
```

### logging.config

Obbligatorio. Specifica il percorso del file di configurazione del logger dedicato. Si consiglia di utilizzare il nome `logback-utility.xml` e fornire questo file come parte dell'applicazione modernizzata. Il modo comune per organizzare questi file consiste nel mettere tutti i file di configurazione dei logger nello stesso posto, di solito nella sottocartella in `/config/logback` cui si `/config` trova la cartella che contiene i file di configurazione yaml. Per ulteriori informazioni, consultate il [Capitolo 3: Configurazione del logback nella documentazione di Logback](#).

```
logging.config : classpath:logback-utility.xml
```

## encoding

Obbligatorio. Specifica il set di caratteri utilizzato dal programma di utilità. Nella maggior parte dei casi, quando si esegue la migrazione da piattaforme z/OS, questo set di caratteri è una variante EBCDIC e deve corrispondere al set di caratteri configurato per le applicazioni modernizzate. L'impostazione predefinita se non è impostata è. ASCII

```
encoding : cp1047
```

## sysPunchEncoding

Facoltativo. Specifica il set di caratteri utilizzato da INFUTILB e DSNUTILB per generare e leggere file SYSPUNCH. Se si utilizzano i file SYSPUNCH della piattaforma precedente così come sono, questo valore dovrebbe essere una variante EBCDIC. L'impostazione predefinita se non è impostata è. ASCII

```
sysPunchEncoding : cp1047
```

## Configurazione dell'origine dati principale

Alcune utilità relative al database, come LOAD e UNLOAD, richiedono l'accesso a un database di destinazione tramite un'origine dati. Come altre definizioni delle fonti di dati incluse in AWS Mainframe Modernization, questo accesso richiede l'utilizzo da parte dell'utente. AWS Secrets Manager Le proprietà che indicano i segreti appropriati in Secrets Manager sono le seguenti:

```
spring.aws.client.datasources.primary.secret
```

Facoltativo. Specifica il segreto in Secrets Manager che contiene le proprietà dell'origine dati.

```
spring.aws.client.datasources.primary.secret: datasource-secret-ARN
```

```
spring.aws.client.datasources.primary.dbname
```

Facoltativo. Specifica il nome del database di destinazione se il nome del database non viene fornito direttamente nel segreto del database, con la proprietà. dbname

```
spring.aws.client.datasources.primary.dbname: target-database-name
```



## treatLargeNumberAsInteger

Facoltativo. Relativo alle specifiche del motore di database Oracle e all'utilizzo delle utilità DSNTEP2/DSNTEP4. Se si imposta questo flag su `true`, i numeri grandi provenienti dal database Oracle (NUMBER (38,0)) vengono trattati come numeri interi. Impostazione predefinita: `false`

```
treatLargeNumberAsInteger : false
```

## ZonedMode

Facoltativo. Imposta la modalità suddivisa in zone per codificare o decodificare i tipi di dati suddivisi in zone. Questa impostazione influenza il modo in cui vengono rappresentate le cifre dei segni. I valori seguenti sono validi:

- `EBCDIC_STRICT`: impostazione predefinita. Usa una definizione rigorosa per la gestione dei segnali. A seconda che il set di caratteri sia EBCDIC o ASCII, la rappresentazione delle cifre dei segni utilizza i seguenti caratteri:
  - Caratteri EBCDIC che corrispondono ai bytes (Cn+Dn) per rappresentare intervalli di cifre positive e negative (to, to). +0 +9 -0 -9 I caratteri vengono visualizzati come { , a, a A I } J R
  - Caratteri ASCII che corrispondono ai bytes (3n+7n) per rappresentare intervalli di cifre positive e negative (+0to+9, -0 to). -9 I caratteri vengono visualizzati come 0 da, a 9 p y
- `EBCDIC_MODIFIED`: utilizza una definizione modificata per la gestione dei segnali. Sia per EBDIC che per ASCII, lo stesso elenco di caratteri rappresenta le cifre dei segni, ovvero da mappato +0 a + a e +9 mappato a { + A a. I -0 -9 } J R \
- `AS400`: da utilizzare per asset legacy modernizzati che provengono da piattaforme iSeries (AS400).

```
zonedMode:EBCDIC_STRICT
```

## jcl.type

Facoltativo. Indica il tipo precedente di script JCL modernizzati. L'utilità IDCAMS utilizza questa impostazione per personalizzare il codice restituito se il JCL che invoca è di tipo. `vse` I valori validi sono:

- `mvs` (predefinito)
- `vse`

```
jcl.type : mvs
```

## Proprietà relative alle utilità Database Unload

Utilizzate queste proprietà per configurare le utilità che scaricano le tabelle del database in set di dati. Tutte le seguenti proprietà sono opzionali.

Questo esempio mostra tutte le possibili proprietà di scarico.

```
# Unload properties
# For date/time: if use database configuration is enabled, formats are ignored
# For nbi; use hexadecimal syntaxe to specify the byte value
unload:
sqlCodePointShift: 0
nbi:
whenNull: "6F"
whenNotNull: "00"
useDatabaseConfiguration: false
format:
date: MM/dd/yyyy
time: HH.mm.ss
timestamp: yyyy-MM-dd-HH.mm.ss.SSSSSS
chunkSize: 0
fetchSize: 0
varCharIsNull: false
columnFiller: space
```

### sqlCodePointShift

Facoltativo. Specifica un valore intero che rappresenta lo spostamento di punti del codice SQL utilizzato nei dati. Il valore predefinito è 0. Ciò significa che non viene effettuato alcun cambio di punto di codice. Allinea questa impostazione al parametro SQL code point shift utilizzato per le applicazioni modernizzate. Quando è in uso lo spostamento dei punti di codice, il valore più comune per questo parametro è 384.

```
unload.sqlCodePointShift: 0
```

### nbi

Facoltativo. Specifica un byte di indicatore nullo. Si tratta di un valore esadecimale (come stringa) aggiunto a destra del valore del dato. I due valori possibili sono i seguenti:

- WhenNull: aggiunge il valore esadecimale quando il valore dei dati è nullo. Il valore predefinito è 6`. A volte viene invece utilizzato il valore alto. FF

```
unload.nbi.whenNull: "6F"
```

- `whenNotNull`: aggiungi il valore esadecimale quando il valore dei dati non è nullo, ma la colonna è annullabile. L'impostazione predefinita è (valore basso). `00`

```
unload.nbi.whenNotNull: "00"
```

## useDatabaseConfiguration

Facoltativo. Specifica le proprietà di formattazione di data e ora. Viene utilizzato per gestire oggetti data/ora nelle query UNLOAD. Il valore predefinito è `false`.

- Se impostato su `true`, utilizza le `pgmTimestampFormat` proprietà `pgmDateFormat` `pgmTimeFormat`, e del file di configurazione principale (`application-main.yml`)
- Se impostato su `false`, utilizza le seguenti proprietà di formattazione di data e ora:
  - `unload.format.date`: specifica un modello di formattazione della data. Il valore predefinito è `MM/dd/yyyy`.
  - `unload.format.time`: specifica un modello di formattazione dell'ora. Il valore predefinito è `HH.mm.ss`.
  - `unload.format.timestamp`: specifica un modello di formattazione del timestamp. Il valore predefinito è `yyyy-MM-dd-HH.mm.ss.SSSSSS`.

## ChunkSize

Facoltativo. Specifica la dimensione dei blocchi di dati utilizzati per creare set di dati SYSREC. Questi set di dati sono l'obiettivo dell'operazione di scaricamento del set di dati, con operazioni parallele. L'impostazione predefinita è `0` (nessun blocco).

```
unload.chunkSize:0
```

## FetchSize

Facoltativo. Specifica la dimensione di recupero dei dati. Il valore è il numero di record da recuperare contemporaneamente quando viene utilizzata una strategia di blocchi di dati. Default: `0`.

```
unload.fetchSize:0
```

## varCharIsNullo

Facoltativo. Specifica come gestire una colonna varchar non annullabile con contenuto vuoto. Il valore predefinito è `false`.

Se si imposta questo valore su `true`, il contenuto della colonna viene trattato come una stringa vuota ai fini dello scaricamento, anziché come una singola stringa di spazio. Imposta questo flag solo `true` per il case del motore di database Oracle.

```
unload.varCharIsNull: false
```

## ColumnFiller

Facoltativo. Specifica il valore da utilizzare per riempire le colonne scariche nelle colonne varchar. I valori possibili sono spazi o valori bassi. L'impostazione predefinita è lo spazio.

```
unload.columnFiller: space
```

## Proprietà relative al caricamento del database

Utilizzate queste proprietà per configurare le utilità che caricano i record dei set di dati in un database di destinazione, ad esempio DSNUTILB. Tutte le seguenti proprietà sono facoltative.

Questo esempio mostra tutte le possibili proprietà di carico.

```
# Load properties
# Batch size for DSNUTILB Load Task
load:
sqlCodePointShift: 384
batchSize: 500
format:
localDate: dd.MM.yyyy|dd/MM/yyyy|yyyy-MM-dd
dbDate: yyyy-MM-dd
localTime: HH:mm:ss|HH.mm.ss
dbTime: HH:mm:ss

table-mappings:
TABLE_1_NAME : LEGACY_TABLE_1_NAME
TABLE_2_NAME : LEGACY_TABLE_2_NAME
```

## sqlCodePointShift

Facoltativo. Specifica un valore intero che rappresenta lo spostamento di punti del codice SQL utilizzato nei dati. Il valore predefinito è 0, il che significa che le applicazioni non effettuano lo spostamento dei punti di codice. Allinea questa impostazione al parametro SQL code point shift utilizzato per le applicazioni modernizzate. Quando si utilizzano gli spostamenti di punti di codice, il valore più comune per questo parametro è 384.

```
load.sqlCodePointShift : 384
```

## batchSize

Facoltativo. Specifica un valore intero che rappresenta il numero di record da trattare prima di inviare un'istruzione batch effettiva al database. Il valore predefinito è 0.

```
load.batchSize: 500
```

## format

Facoltativo. Specifica i modelli di formattazione di data e ora da utilizzare per le conversioni di data/ora durante le operazioni di caricamento del database.

- `load.format.localDate`: modello di formattazione locale della data. L'impostazione predefinita di questa opzione è `dd.MM.yyyy | dd/MM/yyyy | yyyy-MM-dd`.
- `load.format.dbDate`: modello di formattazione della data del database. L'impostazione predefinita di questa opzione è `yyyy-MM-dd`.
- `load.format.localTime`: modello di formattazione dell'ora locale. L'impostazione predefinita di questa opzione è `HH:mm:ss | HH.mm.ss`.
- `load.format.dbTime`: modello di formattazione dell'ora del database. L'impostazione predefinita di questa opzione è `HH:mm:ss`.

## mappature di tabelle

Facoltativo. Specifica una raccolta di mappature fornite dal cliente tra nomi di tabelle precedenti e moderni. Il programma di utilità DSNUTILB utilizza queste mappature.

Specificate i valori nel seguente formato: `MODERN_TABLE_NAME: LEGACY_TABLE_NAME`

Ecco un esempio:

```
table-mappings:
  TABLE_1_NAME : LEGACY_TABLE_1_NAME
  TABLE_2_NAME : LEGACY_TABLE_2_NAME
  ...
  TABLE_*N*_NAME : LEGACY_TABLE_*N*_NAME
```

### Note

All'avvio dell'applicazione di utilità, registra in modo esplicito tutte le mappature fornite.

## Aggiungi proprietà di configurazione per il motore AWS Blu Age

Puoi aggiungere un file nella config cartella della tua applicazione refactorizzata che ti darà accesso alle nuove funzionalità del motore di runtime Blu Age. AWS È necessario assegnare un nome a questo file. `user-properties.yml` Questo file non sostituisce la definizione dell'applicazione ma la estende. Questo argomento descrive le proprietà che è possibile includere nel `user-properties.yml` file.

### Note

Non è possibile modificare alcuni parametri perché sono controllati dalla modernizzazione del AWS mainframe o dalla definizione dell'applicazione. Tutti i parametri definiti nella definizione dell'applicazione hanno la priorità sui parametri specificati. `user-properties.yml`

Per ulteriori informazioni sulla struttura delle applicazioni rifattorizzate, vedere. [Struttura delle applicazioni gestite da AWS Blu Age](#)

Il diagramma seguente mostra dove collocare il `user-properties.yml` file all'interno della struttura dell'applicazione di esempio AWS Blu Age, `PlanetsDemo`

```
PlanetsDemo-v1/
  ## config/
  # ## application-PlanetsDemo.yml
  # ## user-properties.yml
  ## jics/
  ## webapps/
```

## Riferimento alle proprietà di configurazione

Questo è l'elenco delle proprietà disponibili. Tutti i parametri sono opzionali:

### Argomenti

- [Proprietà dell'applicazione Gapwalk](#)
- [Proprietà batchscript di Gapwalk](#)
- [Proprietà Gapwalk Blugen](#)
- [Proprietà del comando Gapwalk CL](#)
- [Proprietà del runner Gapwalk CL](#)
- [Proprietà Gapwalk JHDB](#)
- [Proprietà Gapwalk JICS](#)
- [Proprietà di runtime di Gapwalk](#)
- [Proprietà del programma di utilità Gapwalk](#)
- [Altre proprietà](#)

### Proprietà dell'applicazione Gapwalk

#### bluesam.FileLoading.CommitInterval

Facoltativo. L'intervallo di commit di bluesam.

Tipo: numero

Valore predefinito: 100000

#### card.encoding

Facoltativo. Codifica delle carte: da utilizzare con. useControlMVariable

Tipo: stringa

Predefinito: CP1145

#### controlla la dimensione del file di input

Facoltativo. Specifica se rilasciare un segno di spunta se la dimensione del file è un multiplo della dimensione del record.

Tipo: booleano

Impostazione predefinita: false

`database.cursor.overflow.allowed`

Facoltativo. Specifica se consentire l'overflow del cursore. Imposta per `true` eseguire una chiamata successiva sul cursore indipendentemente dalla sua posizione. Imposta `false` per verificare se il cursore si trova nell'ultima posizione prima di eseguire una chiamata successiva sul cursore. Abilita solo se il cursore è SCORREVOLE (SENSIBILE o INSENSIBILE)

Tipo: booleano

Impostazione predefinita: true

Semplificatore di dati. `onInvalidNumericData`

Facoltativo. Come reagire quando si decodificano dati numerici non validi. I valori consentiti sono `reject`, `toleratespaces`, `toleratespaceslowvalues` e `toleratemo`

Tipo: stringa

Impostazione predefinita: rifiutare

`defaultKeepExistingFile`

Facoltativo. Specifica se impostare il valore precedente predefinito del set di dati.

Tipo: booleano

Impostazione predefinita: false

`disposition.checkexistence`

Facoltativo. Specifica se rilasciare un check on file existence per Dataset with DISP SHR o OLD.

Tipo: booleano

Impostazione predefinita: false

`ExternalSort.Threshold`

Facoltativo. La soglia di ordinamento: quando passare all'ordinamento esterno (unione).

Tipo: stringa

Di default: null

`externalSort.threshold: 12MB`



## ForceHR

Facoltativo. Specifica se utilizzare Human Readable SYSPRINT, sulla console o sull'output del file.

Tipo: booleano

Impostazione predefinita: false

## Data forzata

Facoltativo. Impone una data e un'ora specifiche nel database. Utilizzare solo durante lo sviluppo e il test.

Di default: null

`forcedDate: 2022-08-26T12:59:58.123456+01:57`

## FrozenDate

Facoltativo. Blocca la data e l'ora nel database. Utilizzare solo durante lo sviluppo e il test.

Impostazione predefinita: false

`frozenDate: false`

## ims.messages.ExtendedSize

Facoltativo. Specifica se impostare ExtendedSize sui messaggi ims.

Tipo: booleano

Impostazione predefinita: false

## LockTimeout

Facoltativo. Il timeout in millisecondi di una transazione quando non è possibile acquisire un blocco entro un periodo di tempo specificato.

Tipo: numero

Impostazione predefinita: 500

## MapTransfo.prefixes

Facoltativo. Elenco di prefissi da utilizzare per la trasformazione delle variabili ControlM. Ciascuno separato da una virgola.

Tipo: stringa

Predefinito: &, @,%%

interrogazione. useConcatCondition

Facoltativo. Specifica se la condizione della chiave è creata dalla concatenazione di chiavi o meno.

Tipo: booleano

Impostazione predefinita: false

RollbackOnRTE

Facoltativo. Specifica se eseguire il rollback della transazione implicita dell'unità di esecuzione sulle eccezioni di runtime.

Tipo: booleano

Impostazione predefinita: false

sctThreadLimit

Facoltativo. Il limite dei thread per l'attivazione degli script.

Tipo: numero

Impostazione predefinita: 5

sqlCodePointSpostamento

Facoltativo. Lo spostamento dei punti del codice sql. Sposta il codepoint per i caratteri di controllo che potremmo incontrare durante la migrazione dei dati rdbms legacy a un rdbms moderno. Ad esempio, è possibile specificare che corrisponda al carattere unicode. 384 \u0180

Tipo: numero

Impostazione Predefinita: 0

sqlIntegerOverflowConsentito

Facoltativo. Specifica se consentire l'overflow di numeri interi SQL, ovvero se è consentito inserire valori più grandi nella variabile host.

Tipo: booleano

Impostazione predefinita: false

stepFailWhenUna curva

Facoltativo. Specifica se alzare unabend se un passaggio fallisce o completa l'esecuzione.

Tipo: booleano

Impostazione predefinita: true

stopExecutionWhenProgNotFound

Facoltativo. Specifica se interrompere l'esecuzione se non viene trovato un programma. Se impostato su true, interrompe l'esecuzione se non viene trovato un programma.

Tipo: booleano

Impostazione predefinita: true

uppercaseUserInput

Facoltativo. Specifica se l'input dell'utente deve essere in lettere maiuscole.

Tipo: booleano

Impostazione predefinita: true

UseControlM Variable

Facoltativo. Specifica se utilizzare la specifica Control-M per la sostituzione delle variabili.

Tipo: booleano

Impostazione predefinita: false

Proprietà batchscript di Gapwalk

encoding

Facoltativo. La codifica utilizzata nei progetti batchscript (non con groovy). Prevede una codifica valida, ,... CP1047 IBM930 ASCII UTF-8

Tipo: stringa

Impostazione predefinita: ASCII

## Proprietà Gapwalk Blugen

### managers.trancode

Facoltativo. La mappatura del trancode del gestore di dialoghi. Consente di mappare un codice di transazione JICS a un gestore di dialoghi. Il formato previsto è `trancode1:dialogManager1;trancode2:dialogManager2;`.

Tipo: stringa

Di default: null

`managers.trancode: OR12:MYDIALOG1`

## Proprietà del comando Gapwalk CL

### comandi disattivati

Facoltativo. Elenco di comandi da disattivare, separati da una virgola. I valori consentiti sono `PGM_BASIC`

`RCVMSGSNDRCVF,CHGVAR,QCLRDTAQ,RTVJOB,ADDFM,ADDPFM,RCVF,OVRDBF,DLTOVR,CPYF,SNDDTAQ`

Utile quando si desidera disabilitare o sovrascrivere un programma esistente. `PGM_BASIC` è un programma di velocità specifico progettato per scopi di debug.

Tipo: stringa

Di default: null

### spring.datasource.primary.jndi-name

Facoltativo. La fonte di dati principale di Java Naming And Directory Interface (jndi).

Tipo: stringa

Predefinito: `jdbc/primary`

### Modalità suddivisa in zone

Facoltativo. La modalità per la codifica o la decodifica dei tipi di dati suddivisi in zone. I valori consentiti sono `//EBCDIC_STRICT`, `EBCDIC_MODIFIED`, `AS400`

Tipo: stringa

Impostazione predefinita: `EBCDIC_STRICT`

## Proprietà del runner Gapwalk CL

### cl.configuration.context.encoding

Facoltativo. La codifica dei file CL. Prevede una codifica valida CP1047,,IBM930... ASCII UTF-8

Tipo: stringa

Predefinito: CP297

### cl.ZonedMode

Facoltativo. La modalità per la codifica o la decodifica dei comandi del linguaggio di controllo (CL). I valori consentiti sono EBCDIC\_STRICT//EBCDIC\_MODIFIED. AS400

Tipo: stringa

Impostazione predefinita: EBCDIC\_STRICT

## Proprietà Gapwalk JHDB

### ims.programs

Facoltativo. Elenco dei programmi IMS da utilizzare. Separare ogni parametro con un punto e virgola (;) e ogni transazione con una virgola (,). Ad esempio: `ims.programs: PCP008,PCT008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;`

Tipo: stringa

Di default: null

### jhdb.checkpointPath

Facoltativo. In caso contrario `jhdb.checkpointPersistence`, questo parametro consente di impostare il percorso di persistenza del checkpoint (posizione di archiviazione del file `checkpoint.dat`), tutti i dati dei checkpoint contenuti nel registro vengono serializzati e sottoposti a backup in un file (`checkpoint.dat`) situato nella cartella fornita. **Nota** che solo i dati del checkpoint (`ScriptID`, `StepID`, posizione del database e area del checkpoint) sono interessati da questo backup.

Tipo: stringa

Predefinito: file:./setup/

## jhdb.CheckPointPersistence

Facoltativo. La modalità di persistenza del checkpoint. I valori consentiti sono `none//add. end`. Si usa `add` per mantenere i checkpoint quando ne viene creato uno nuovo e aggiunto al registro. Utilizzato `end` per mantenere il checkpoint alla chiusura del server. Qualsiasi altro valore disabilita la persistenza. Tieni presente che ogni volta che viene aggiunto un nuovo checkpoint al registro, tutti i checkpoint esistenti verranno serializzati e il file verrà cancellato. Non è un'aggiunta ai dati esistenti nel file. Quindi, a seconda del numero di checkpoint, può avere qualche effetto sulle prestazioni.

Tipo: stringa

Impostazione predefinita: none

## jhdb.configuration.context.encoding

Facoltativo. La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida, `CP1047 IBM930 ASCII UTF-8`

Tipo: stringa

Predefinito: CP297

## jhdb.identificationCardData

Facoltativo. Utilizzato per codificare alcuni «dati della carta di identificazione dell'operatore» nel campo MID indicato dal parametro `CARD`.

Tipo: stringa

Default: ""

## jhdb.lterm

Facoltativo. Consentono di forzare un ID di terminale logico comune nel caso di un'emulazione IMS. Se non è impostato, viene utilizzato `SessionID`.

Tipo: stringa


Di default: null

## jhdb.metadata.extrapath

Un parametro di configurazione che specifica una cartella principale aggiuntiva, specifica per il runtime, per le cartelle `psbs` e `dbds`.

Tipo: stringa

Predefinito: file: ./setup/

 Note

Attualmente, per i vincoli di distribuzione, è necessario copiare le directory dbds e psbs nella directory config dell'applicazione o in una sottodirectory della directory config: ad esempio, config/setup

```
config
|- setup
  |- dbds
  |- psbs
```

e impostate in application-jhdb.yml

```
jhdb.metadata.extrapath: file: ./config/setup/
```

jhdb.navigation.cachenexts

Facoltativo. La durata della cache (in millisecondi) utilizzata nella navigazione gerarchica per un RDBMS.

Tipo: numero

Impostazione predefinita: 5000

jhdb.query.limitJoinUsage

Facoltativo. Specifica se utilizzare il parametro limit join usage sui grafici RDBMS.

Tipo: booleano

Impostazione predefinita: true

jhdb.use-db-prefix

Facoltativo. Specifica se abilitare un prefisso di database nella navigazione gerarchica per un RDBMS.

Tipo: booleano

Impostazione predefinita: true

## Proprietà Gapwalk JICS

### `jics.data.dataJsonInitUbicazione`

Facoltativo. Posizione del file json preparato dall'Analyzer dall'analisi di CSD e utilizzato per inizializzare il database jics,

Tipo: stringa

Default: ""

### `jics.db.dataScriptLocation`

Facoltativo. Posizione dello script `initJics.sql`, preparato da Analyzer dall'analisi delle esportazioni CSD dal mainframe.

Tipo: stringa

Default: ""

### `jics.db.dataTestQueryUbicazione`

Facoltativo. Posizione di uno script sql contenente una singola query sql che dovrebbe restituire un conteggio di oggetti (ad esempio: conteggio del numero di record nella tabella del programma jics). Se il conteggio è uguale a 0, il database verrà caricato utilizzando `jics.db.dataScriptLocation` lo script, altrimenti il caricamento del database verrà saltato.

Tipo: stringa

Default: ""

### `jics.db.ddlScriptLocation`

Facoltativo. La posizione dello script Jics ddl. Consente di avviare lo schema del database jics utilizzando uno script.sql.

Tipo: stringa

Default: ""

`jics.db.ddlScriptLocation: ./jics/sql/jics.sql`



### jics.db.schemaTestQueryUbicazione

Facoltativo. Posizione del file sql che dovrebbe contenere una query univoca che restituisce il numero di oggetti nello schema jics (se presente).

Tipo: stringa

Default: ""

### jics.runUnitLauncherPool.Abilita

Facoltativo. Specifica se attivare il run unit launcher pool in JICS.

Tipo: booleano

Impostazione predefinita: false

### jics.runUnitLauncherDimensioni della piscina

Facoltativo. La dimensione del pool di avvio dell'unità in JICS.

Tipo: numero

Impostazione predefinita: 20

### jics.runUnitLauncherpool.ValidationInterval

Facoltativo: l'intervallo di convalida del pool di avvio dell'unità di esecuzione in JICS, espresso in millisecondi.

Tipo: numero

Impostazione predefinita: 1000

### jics.queues.sqs.region

Facoltativo. Il Regione AWS per Amazon SQS, utilizzato in JICS. Si consiglia di impostare la stessa regione dell'applicazione distribuita per le prestazioni, ma non è obbligatorio.

Tipo: stringa

Impostazione predefinita: eu-west-1

### jics.xa.agent.timeout

Facoltativo. Definisce la durata massima del completamento delle operazioni da parte dell'agente xa responsabile della gestione delle transazioni distribuite.

Tipo: numero

Di default: null

`mq.queues.sqs.region`

Facoltativo. Il Regione AWS per il servizio Amazon SQS MQ.

Tipo: stringa

Impostazione predefinita: eu-west-3

`TaskExecutor.allowCoreThreadTimeOut`

Facoltativo. Specifica se consentire il timeout dei thread principali in JCIS. Ciò consente l'aumento e la riduzione dinamici anche in combinazione con una coda diversa da zero (poiché la dimensione massima del pool aumenterà solo quando la coda sarà piena).

Tipo: booleano

Impostazione predefinita: false

`TaskExecutor.corePoolSize`

Facoltativo. Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per configurare la dimensione del pool principale.

Tipo: numero

Impostazione predefinita: 5

`TaskExecutor.maxPoolSize`

Facoltativo. Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per impostare la dimensione massima del pool (numero massimo di thread paralleli).

Tipo: numero

Impostazione predefinita: 10

`TaskExecutor.QueueCapacity`

Facoltativo. Quando una transazione in un terminale viene avviata tramite uno script groovy, viene creato un nuovo thread. Utilizzate questo parametro per impostare la dimensione

della coda. (= numero massimo di transazioni in sospeso quando viene raggiunto)

`taskExecutor.maxPoolSize`

Tipo: numero

Impostazione predefinita: 50

## Proprietà di runtime di Gapwalk

### Metadati della cache

Facoltativo. Specifica se memorizzare nella cache i metadati del database.

Tipo: booleano

Impostazione predefinita: true

### check-groovy-file

Facoltativo. Specifica se controllare il contenuto dei file Groovy prima della registrazione.

Tipo: booleano

Impostazione predefinita: true

### Statistiche del database

Facoltativo. Specifica se consentire ai costruttori di SQL di raccogliere e visualizzare informazioni statistiche.

Tipo: booleano

Impostazione predefinita: false

### dateTimeFormat

Facoltativo. `dateTimeFormat` Descrive come trasferire la data, l'ora e il tipo di timestamp del database in entità semplificatrici di dati. I valori consentiti sono `ISO``EUR``US``LOCAL`

Tipo: stringa

Impostazione predefinita: ISO

### dbDateFormat

Facoltativo. Il formato della data di destinazione del database.

Tipo: stringa

Predefinito: yyyy-MM-dd

#### dbTimeFormat

Facoltativo. Il formato dell'ora di destinazione del database.

Tipo: stringa

Predefinito: HH:mm:ss

#### dbTimestampFormat

Facoltativo. Il formato del timestamp di destinazione del database.

Tipo: stringa

Predefinito: yyyy-mm-dd HH:mm:ss.ssssss

#### Recupera dimensione

Facoltativo. Il valore fetchSize per i cursori. Da utilizzare quando si recuperano dati utilizzando blocchi tramite utilità load/unload.

Tipo: numero

Impostazione predefinita: 10

#### ForceDisableSQL TrimStringType

Facoltativo. Specifica se disabilitare il taglio di tutti i parametri delle stringhe sql.

Tipo: booleano

Impostazione predefinita: false

#### localDateFormat

Facoltativo. Elenco dei formati di data locali. Separa ogni formato con |.

Tipo: stringa

#### localTimeFormat

Facoltativo. Elenco dei formati dell'ora locale. Separa ogni formato con |.

Tipo: stringa

localTimestampFormat

Facoltativo. Elenco dei formati di timestamp locali. Separa ogni formato con. |

Tipo: stringa

Impostazione predefinita:

pgmDateFormat

Facoltativo. Il formato di data e ora utilizzato nei programmi.

Tipo: stringa

Predefinito: yyyy-MM-dd

pgmTimeFormat

Facoltativo. Il formato dell'ora usato per l'esecuzione di pgm (programmi).

Tipo: stringa

Predefinito: hh.mm.ss

pgmTimestampFormat

Facoltativo. Il formato del timestamp.

Tipo: stringa

Valore predefinito: yyyy-MM-DD-HH.MM.ss.ssssss

Proprietà del programma di utilità Gapwalk

jcl.type

Facoltativo. .jcl tipo di file. I valori consentiti sono jcl/vse. I comandi PRINT/REPRO dell'utilità IDCAMS restituiscono 4 se il file è vuoto per jcl non vse.

Tipo: stringa

Impostazione predefinita: mvs

**listcat.variablelengthpreprocessor.enabled**

Facoltativo. Specifica se abilitare il preprocessore a lunghezza variabile per il comando LISTCAT.

Tipo: booleano

Impostazione predefinita: false

**listcat.variablelengthpreprocessor.type**

Facoltativo. Il tipo di oggetti contenuti nel file listcat, se abilitato.

`listcat.variablelengthpreprocessor.enabled` I valori consentiti sono rdw/bdw.

Tipo: stringa

Predefinito: rdw

**load.batchSize**

Facoltativo. La dimensione del batch dell'utilità di caricamento.

Tipo: numero

Impostazione Predefinita: 0

**load.format.dbdate**

Facoltativo. Il formato del database dell'utilità di caricamento da utilizzare.

Tipo: stringa

Predefinito: yyyy-MM-dd

**Load.format.dbTime**

Facoltativo. Il tempo di utilizzo del database dell'utilità di caricamento.

Tipo: stringa

Impostazione predefinita: HH:mm:ss

**load.format.localDate**

Facoltativo. Il formato di data locale dell'utilità di caricamento da utilizzare.

Tipo: stringa

Impostazione predefinita: `gd.mm.yyyy|gg/mm/yyyy|yyyy-MM-DD`

`load.format.localTime`

Facoltativo. Il formato dell'ora locale dell'utilità di caricamento da utilizzare.

Tipo: stringa

Predefinito: `HH:mm:ss|Hh.mm.ss`

`caricare.sqlCodePointCambio`

Facoltativo. L'utilità `pointshift` del codice SQL per il caricamento. Esegue il processo di spostamento dei caratteri. Richiesto quando il database di destinazione di DB2 è PostgreSQL.

Tipo: numero

Impostazione Predefinita: 0

`sysPunchEncoding`

Facoltativo. Il set di caratteri di codifica `syspunch`. I valori supportati sono/. `Cp1047 ASCII`

Tipo: stringa

Impostazione predefinita: `ASCII`

`treatLargeNumberAsInteger`

Facoltativo. Specifica se trattare i numeri grandi come `Integer`. Per impostazione predefinita, vengono `BigDecimal` trattati come tali.

Tipo: booleano

Impostazione predefinita: `false`

`unload.chunksize`

Facoltativo. Dimensione del blocco utilizzata per l'utilità di scarico.

Tipo: numero

Impostazione Predefinita: 0

`unload.columnFiller`

Facoltativo. Il riempitore di colonne dell'utilità di scarico.

Tipo: stringa

Predefinito: spazio

`unload.fetchSize`

Facoltativo. Consente di regolare la dimensione del recupero quando si gestiscono i cursori nell'utilità `unload`.

Tipo: numero

Impostazione Predefinita: 0

`unload.format.date`

Facoltativo. Se abilitato, `unload.useDatabaseConfiguration` il formato della data da utilizzare nell'utilità di scarico. Per ulteriori informazioni, vedete [unload.format.date](#).

Tipo: stringa

Impostazione predefinita: MM/gg/aaaa

`unload.format.time`

Facoltativo. Se abilitato, `unload.useDatabaseConfiguration` il formato dell'ora da utilizzare nell'utilità di scarico.

Tipo: stringa

Predefinito: hh.mm.ss

`unload.format.timestamp`

Facoltativo. Se abilitato, `unload.useDatabaseConfiguration` il formato del timestamp da utilizzare nell'utilità `unload`.

Tipo: stringa

Valore predefinito: yyyy-MM-DD-HH.MM.ss.ssssss

`scarica.nbi.whenNotNull`

Facoltativo. Il valore Null Byte Indicator (nbi) da aggiungere quando il valore del database non è nullo.

Tipo: esadecimale



Valore predefinito: 00

`unload.nbi.whenNull`

Facoltativo. Il valore dell'indicatore Null Byte (nbi) da aggiungere quando il valore del database è nullo.

Tipo: esadecimale

Predefinito: 6F

`scarica.nbi.writeNullIndicator`

Facoltativo. Specifica se scrivere l'indicatore null nel file di output di scaricamento.

Tipo: booleano

Impostazione predefinita: false

`scaricare.sqlCodePointCambio`

Facoltativo. L'utilità pointshift del codice SQL per unload. Esegue il processo di cambio dei caratteri. Richiesto quando il database di destinazione di DB2 è Postgresql.

Tipo: numero

Impostazione Predefinita: 0

`scaricare.useDatabaseConfiguration`

Facoltativo. Specifica se utilizzare la configurazione di data o ora da `application-main.yml` nell'utilità unload.

Tipo: booleano

Impostazione predefinita: false

`scaricare.varCharIsNullo`

Facoltativo. Usa questo parametro nel programma INFTILB, se impostato su `true` allora tutti i campi non annullabili con valori vuoti (spazi) restituiscono una stringa vuota.

Tipo: booleano

Impostazione predefinita: false

## Altre proprietà

### qtemp.cleanup.threshold. hours

Facoltativo. Per specificare quando è abilitato. `qtemp.dblog` La durata della partizione db (in ore).

Tipo: numero

Impostazione Predefinita: 0

### qtemp.dblog

Facoltativo. Se abilitare la registrazione del database QTEMP.

Tipo: booleano

Impostazione predefinita: false

### qtemp.uuid.length

Facoltativo. La lunghezza dell'ID univoco di QTEMP.

Tipo: numero

Valore predefinito: 9

### quartz.scheduler. stand-by-if-error

Facoltativo. Specifica se attivare l'esecuzione del lavoro se lo scheduler del lavoro è in modalità standby. Se vero, quando è abilitata l'esecuzione del lavoro non viene attivata.

Tipo: booleano

Impostazione predefinita: false

### warmUpCache

Facoltativo. Specifica se caricare tutti i dati della tabella Datacom in una cache di riscaldamento all'avvio del server.

Tipo: booleano

Impostazione predefinita: false

# AWS Riferimento alla definizione delle applicazioni di modernizzazione del mainframe

In AWS Mainframe Modernization, è possibile configurare le applicazioni mainframe migrate in un file JSON di definizione dell'applicazione, specifico per il motore di runtime scelto. Una definizione di applicazione contiene sia informazioni generali che informazioni specifiche del motore. Questo argomento descrive le definizioni delle applicazioni AWS Blu Age e Micro Focus e identifica tutti gli elementi obbligatori e opzionali.

## Argomenti

- [Sezione generale dell'intestazione](#)
- [Panoramica della sezione relativa alle definizioni](#)
- [AWS Esempio di definizione dell'applicazione Blu Age](#)
- [AWS Dettagli sulla definizione di Blu Age](#)
- [Definizione dell'applicazione Micro Focus](#)
- [Dettagli sulla definizione di Micro Focus](#)

## Sezione generale dell'intestazione

Ogni definizione di applicazione inizia con informazioni generali sulla versione del modello e sulle posizioni di origine. La versione corrente della definizione dell'applicazione è 2.0. Sebbene la versione 1 funzioni ancora, si trova in un percorso obsoleto. Ti consigliamo di utilizzare la versione 2 quando crei o aggiorni applicazioni.

Utilizzate la seguente struttura per specificare la versione del modello e le posizioni di origine.

```
"template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ]
```

]

**Note**

Puoi usare la seguente sintassi se vuoi inserire S3 ARN come bucket s3:

```
"template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "arn:aws:s3:::mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ]
```

**versione modello**

Obbligatorio. Specifica la versione del file di definizione dell'applicazione. Non modificare questo valore. Attualmente, l'unico valore consentito è 2.0. Specificare `template-version` con una stringa.

**ubicazioni di origine**

Specifica le posizioni dei file e delle altre risorse richieste dall'applicazione durante l'esecuzione.

**properties**

Fornisce i dettagli della posizione di origine. Ogni proprietà viene specificata con una stringa.

- `s3-bucket`- Obbligatorio. Specifica il nome del bucket Amazon S3 in cui sono archiviati i file.
- `s3-key-prefix`- Obbligatorio. Specifica il nome della cartella nel bucket Amazon S3 in cui sono archiviati i file.

**Note**

Assicurati di specificare il nome del bucket Amazon S3, non l'ARN del bucket. Non specificare un percorso assoluto per le risorse nel bucket.

## Panoramica della sezione relativa alle definizioni

Specifica le definizioni delle risorse dei servizi, delle impostazioni, dei dati e di altre risorse tipiche necessarie per l'esecuzione dell'applicazione. Quando si aggiorna la definizione di un'applicazione, AWS Mainframe Modernization rileva le modifiche confrontando `definition` gli elenchi `source-locations` e delle versioni precedenti e correnti del file JSON di definizione dell'applicazione.

La sezione delle definizioni è specifica del motore e soggetta a modifiche. Le sezioni seguenti mostrano esempi di definizioni di applicazioni specifiche del motore per entrambi i motori.

### AWS Esempio di definizione dell'applicazione Blu Age

```
{
  "template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ],
  "definition" : {
    "listeners": [{
      "port": 8194,
      "type": "http"
    }],
    "ba-application": {
      "app-location": "${s3-source}/murachs-v6/"
    },
    "blusam": {
      "db": {
        "nb-threads": 8,
        "batch-size": 10000,
        "name": "blusam",
        "secret-manager-arn": "arn:aws:secretsmanager:us-west-2:111122223333:secret:blusam-FfmXLG"
      },
      "redis": {
        "hostname": "blusam.c3geul.ng.0001.usw2.cache.amazonaws.com",
```

```

        "port": 6379,
        "useSsl": true,
        "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:bluesamredis-nioefm"
    }
}
}
}

```

## AWS Dettagli sulla definizione di Blu Age

### Ascoltatori: obbligatorio

Specificate la porta che utilizzerete per accedere all'applicazione tramite Elastic Load Balancing creato da AWS Mainframe Modernization. Utilizzate la seguente struttura:

```

"listeners": [{
    "port": 8194,
    "type": "http"
}],

```

#### port

Obbligatorio. È possibile utilizzare qualsiasi porta disponibile ad eccezione delle porte note da 0 a 1023. Si consiglia di utilizzare l'intervallo da 8192 a 8199. Assicuratevi che non ci siano altri listener o applicazioni che operano su questa porta.

#### type (tipo)

Obbligatorio. Attualmente è supportato solo http.

## AWS Applicazione Blu Age: richiesta

Specificate la posizione in cui il motore preleva il file di immagine dell'applicazione utilizzando la seguente struttura.

```

"ba-application": {
    "app-location": "${s3-source}/murachs-v6/",
    "files-directory": "/m2/mount/myfolder",
    "enable-jics": <true|false>,

```

```

    "shared-app-location": "${s3-source}/shared/"
  },

```

### posizione dell'app

La posizione specifica in Amazon S3 in cui è archiviato il file di immagine dell'applicazione.

### file-directory

Facoltativo. La posizione dei file di input/output per i batch. Deve essere una sottocartella della configurazione del punto di montaggio Amazon EFS o Amazon FSx a livello di ambiente.

### abilita jics

Facoltativo. Specifica se abilitare JICS. Il valore predefinito è true. L'impostazione su false impedisce la generazione del database JICS.

### shared-app-location

Facoltativo. Ulteriore posizione in Amazon S3 in cui sono archiviati gli elementi applicativi condivisi. Può contenere lo stesso tipo di struttura applicativa di app-location.

## BluSAM - opzionale

Specificate il database BluSAM e la cache Redis utilizzando la seguente struttura.

```

"blusam": {
  "db": {
    "nb-threads": 8,
    "batch-size": 10000,
    "name": "blusam",
    "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:blusam-FfmXLG"
  },
  "redis": {
    "hostname": "blusam.c3geul.ng.0001.usw2.cache.amazonaws.com",
    "port": 6379,
    "useSsl": true,
    "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:bluesamredis-nioefm"
  }
}

```

## db

Speciifica le proprietà del database utilizzato con l'applicazione. Il database deve essere un database Aurora PostgreSQL. È possibile specificare le seguenti proprietà:

- `nb-threads`- Facoltativo. Specifica quanti thread dedicati vengono utilizzati per il meccanismo di scrittura su cui si basa il motore blusam. Il valore di default è 8.
- `batch-size`- Facoltativo. Speciifica la soglia utilizzata dal meccanismo write-behind per avviare le operazioni di archiviazione in batch. La soglia rappresenta il numero di record modificati che avvieranno un'operazione di archiviazione in batch per garantire che i record modificati vengano mantenuti. Il trigger stesso si basa su una combinazione delle dimensioni del batch e del tempo trascorso di un secondo, a seconda di quale dei due valori viene raggiunto per primo. Il valore predefinito è 10000.
- `name`- Facoltativo. Specificate il nome del database.
- `secret-manager-arn`- Speciifica l'Amazon Resource Name (ARN) del segreto che contiene le credenziali del database. Per ulteriori informazioni, consulta [Passaggio 4: Creare e configurare un AWS Secrets Manager database segreto](#).

## redis

Speciifica le proprietà della cache Redis che l'applicazione utilizza per archiviare i dati temporanei necessari in una posizione centrale per migliorare le prestazioni. Si consiglia di crittografare e proteggere con password la cache Redis.

- `hostname`- Specifica la posizione della cache Redis.
- `port`- Speciifica la porta, in genere 6379, in cui la cache Redis invia e riceve le comunicazioni.
- `useSsl`- Speciifica se la cache Redis è crittografata. Se la cache non è crittografata, impostala su `falseuseSsl`.
- `secret-manager-arn`- Speciifica l'Amazon Resource Name (ARN) del segreto che contiene la password della cache Redis. Se la cache Redis non è protetta da password, non specificare `secret-manager-arn`. Per ulteriori informazioni, consulta [Passaggio 4: Creare e configurare un AWS Secrets Manager database segreto](#).

## AWS Code di messaggi Blu Age: facoltative

Specificare i dettagli di connessione JMS-MQ per l'applicazione AWS Blu Age.

```
"message-queues": [  
  {
```



```
    "product-type": "JMS-MQ",
    "queue-manager": "QMgr1",
    "channel": "mqChannel1",
    "hostname": "mqserver-host1",
    "port": 1414,
    "user-id": "app-user1",
    "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  },
  {
    "product-type": "JMS-MQ",
    "queue-manager": "QMgr2",
    "channel": "mqChannel2",
    "hostname": "mqserver-host2",
    "port": 1412,
    "user-id": "app-user2",
    "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  }
]
```

### tipo di prodotto

Obbligatorio. Specifica il tipo di prodotto. Attualmente, questo può essere solo «JMS-MQ» per le applicazioni AWS Blu Age.

### gestore di code

Obbligatorio. Specifica il nome del gestore delle code.

### canale

Obbligatorio. Specifica il nome del canale di connessione al server.

### hostname

Obbligatorio. Specifica il nome host del server della coda dei messaggi.

### port

Obbligatorio. Specifica il numero di porta del listener su cui il server è in ascolto.

### ID utente

Facoltativo. Specifica l'ID dell'account utente autorizzato a eseguire operazioni di coda dei messaggi sul canale specificato.

## secret-manager-arn

Facoltativo. Specifica l'Amazon Resource Name (ARN) di Secrets Manager che fornisce la password dell'utente specificato.

## Definizione dell'applicazione Micro Focus

La seguente sezione di definizione di esempio riguarda il motore di runtime Micro Focus e contiene elementi sia obbligatori che opzionali.

```
{
  "template-version": "2.0",
  "source-locations": [
    {
      "source-id": "s3-source",
      "source-type": "s3",
      "properties": {
        "s3-bucket": "mainframe-deployment-bucket-aaa",
        "s3-key-prefix": "v1"
      }
    }
  ],
  "definition" : {
    "listeners": [{
      "port": 5101,
      "type": "tn3270"
    }],
    "dataset-location": {
      "db-locations": [{
        "name": "Database1",
        "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456"
      }]
    },
    "cognito-auth-handler": {
      "user-pool-id": "cognito-idp.us-west-2.amazonaws.com/us-west-2_rvYFnQIxL",
      "client-id": "58k05jb8grukjjsudm5hhn1v87",
      "identity-pool-id": "us-west-2:64464b12-0bfb-4dea-ab35-5c22c6c245f6"
    },
    "ldap-ad-auth-handler": {
      "ldap-ad-connection-secrets": [LIST OF AD-SECRETS]
    },
    "batch-settings": {
```

```

    "initiators": [{
      "classes": ["A", "B"],
      "description": "initiator...."
    }],
    "jcl-file-location": "${s3-source}/batch/jcl"
  },
  "cics-settings": {
    "binary-file-location": "${s3-source}/cics/binaries",
    "csd-file-location": "${s3-source}/cics/def",
    "system-initialization-table": "BNKCICV"
  },
  "xa-resources" : [{
    "name": "XASQL",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
    "module": "${s3-source}/xa/ESPGSQLXA64.so"
  }]
}

```

## Dettagli sulla definizione di Micro Focus

Il contenuto nella sezione di definizione del file di definizione dell'applicazione Micro Focus varia a seconda delle risorse che l'applicazione mainframe migrata richiede in fase di esecuzione.

### Listener (i): obbligatorio

Specificate un ascoltatore utilizzando la seguente struttura:

```

"listeners": [{
  "port": 5101,
  "type": "tn3270"
}],

```

### port

Per tn3270, l'impostazione predefinita è 5101. Per altri tipi di service listener, la porta varia. È possibile utilizzare qualsiasi porta disponibile ad eccezione delle porte note da 0 a 1023. Ogni listener deve avere una porta distinta. Gli ascoltatori non devono condividere le porte. Per ulteriori informazioni, vedete [Listener Control nella documentazione di Micro Focus Enterprise Server](#).

## type (tipo)

Specifica il tipo di listener di servizio. Per ulteriori informazioni, vedete [Listener](#) nella documentazione di Micro Focus Enterprise Server.

## Posizioni dei set di dati: obbligatorie

Specificate la posizione del set di dati utilizzando la seguente struttura.

```
"dataset-location": {
  "db-locations": [{
    "name": "Database1",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456"
  }],
}
```

### db-locations

Specifica la posizione dei set di dati creati dall'applicazione migrata. Attualmente, AWS Mainframe Modernization supporta solo set di dati provenienti da un singolo database VSAM.

- **name**- Specifica il nome dell'istanza del database che contiene i set di dati creati dall'applicazione migrata.
- **secret-manager-arn**- Specifica l'Amazon Resource Name (ARN) del segreto che contiene le credenziali del database.

## Gestore di autenticazione e autorizzazione Amazon Cognito: opzionale

AWS La modernizzazione del mainframe utilizza Amazon Cognito per l'autenticazione e l'autorizzazione delle applicazioni migrate. Specificare il gestore di autenticazione Amazon Cognito utilizzando la seguente struttura.

```
"cognito-auth-handler": {
  "user-pool-id": "cognito-idp.Region.amazonaws.com/Region_rvYFnQIxL",
  "client-id": "58k05jb8grukjjsudm5hhn1v87",
  "identity-pool-id": "Region:64464b12-0bfb-4dea-ab35-5c22c6c245f6"
}
```

## user-pool-id

Specifica il pool AWS di utenti di Amazon Cognito utilizzato da Mainframe Modernization per autenticare gli utenti dell'applicazione migrata. Il pool Regione AWS per il pool di utenti deve corrispondere a quello dell'applicazione Mainframe Modernization Regione AWS . AWS

## identificativo del cliente

Specifica l'applicazione migrata a cui l'utente autenticato può accedere.

## identity-pool-id

Specifica il pool di identità di Amazon Cognito in cui l'utente autenticato scambia un token del pool di utenti con credenziali che consentono all'utente di accedere alla modernizzazione del mainframe. AWS Il pool di identità Regione AWS for the identity deve corrispondere a quello dell'applicazione Mainframe Modernization Regione AWS . AWS

## Gestore LDAP e Active Directory: opzionale

È possibile integrare l'applicazione con Active Directory (AD) o qualsiasi tipo di server LDAP per consentire agli utenti dell'applicazione di utilizzare le proprie credenziali LDAP/AD per l'autorizzazione e l'autenticazione.

Per integrare la tua applicazione con AD

1. Seguite i passaggi descritti in [Configurazione di Active Directory per Enterprise Server Security](#) nella documentazione di Micro Focus Enterprise Server.
2. Crea un AWS Secrets Manager segreto con i tuoi dettagli AD/LDAP per ogni server AD/LDAP che desideri utilizzare con la tua applicazione. Per informazioni su come creare un segreto, consulta [Create an AWS Secrets Manager secret](#) nella AWS Secrets Manager User Guide. Per il tipo di segreto, scegli Altro tipo di segreto e includi le seguenti coppie chiave-valore.

```
{
  "connectionPath"      : "<HOST-ADDRESS>:<PORT>",
  "authorizedId"        : "<USER-FULL-DN>",
  "password"            : "<PASSWORD>",
  "baseDn"              : "<BASE-FULL-DN>",
  "userClassDn"         : "<USER-TYPE>",
  "userContainerDn"     : "<USER-CONTAINER-DN>",
  "groupContainerDn"    : "<GROUP-CONTAINER-DN>",
  "resourceContainerDn" : "<RESOURCE-CONTAINER-DN>"
}
```

```
}
```

### Consigli sulla sicurezza

- Infatti `connectionPath`, AWS Mainframe Modernization supporta i protocolli LDAP e LDAP over SSL (LDAPS). Si consiglia di utilizzare LDAPS perché è più sicuro e impedisce la visualizzazione delle credenziali nelle trasmissioni di rete.
- Inoltre `password`, ti consigliamo di specificare le credenziali di un utente senza più autorizzazioni rispetto alle autorizzazioni di sola lettura e verifica più restrittive necessarie per l'esecuzione dell'applicazione. `authorizedId`
- Si consiglia di ruotare regolarmente le credenziali AD/LDAP.
- Non creare utenti AD con il nome utente o. `awsuser` `mfuser` Questi due nomi utente sono riservati all' AWS uso.

Di seguito è riportato un esempio.

```
{
  "connectionPath" : "ldaps://msad4.m2.example.people.aws.dev:636",
  "authorizedId" :
  "CN=LDAPUser,OU=Users,OU=msad4,DC=msad4,DC=m2,DC=example,DC=people,DC=aws,DC=dev",
  "password" : "ADPassword",
  "userContainerDn" : "CN=Enterprise Server Users,CN=Micro Focus,CN=Program
Data,OU=msad4,DC=msad4,DC=m2,DC=example,DC=people,DC=aws,DC=dev",
  "groupContainerDn" : "CN=Enterprise Server Groups,CN=Micro Focus,CN=Program
Data,OU=msad4,DC=msad4,DC=m2,DC=example,DC=people,DC=aws,DC=dev",
  "resourceContainerDn" : "CN=Enterprise Server Resources,CN=Micro
Focus,CN=Program Data,OU=msad4,DC=msad4,DC=m2,DC=example,DC=people,DC=aws,DC=dev"
}
```

Crea il segreto con una chiave KMS gestita dal cliente. Devi concedere a AWS Mainframe Modernization le autorizzazioni `GetSecretValue` e le `DescribeSecret` autorizzazioni sulla chiave segreta `Decrypt` e `DescribeKey` le autorizzazioni sulla chiave KMS. Per ulteriori informazioni, consulta [Autorizzazioni per la chiave KMS nella Guida per l'utente](#). AWS Secrets Manager

3. Aggiungi quanto segue alla definizione dell'applicazione.

```
"ldap-ad-auth-handler": {  
  "ldap-ad-connection-secrets": [LIST OF AD/LDAP SECRETS]  
}
```

Di seguito è riportato un esempio.

```
"ldap-ad-auth-handler": {  
  "ldap-ad-connection-secrets": ["arn:aws:secrets:1234:us-east-1:secret:123456"]  
}
```

Il gestore di autenticazione LDAP/AD è disponibile per Micro Focus 8.0.11 e versioni successive.

## Impostazioni Batch: obbligatorie

Specificate i dettagli richiesti dai processi batch eseguiti come parte dell'applicazione utilizzando la seguente struttura.

```
"batch-settings": {  
  "initiators": [{  
    "classes": ["A", "B"],  
    "description": "initiator...."  
  }],  
  "jcl-file-location": "${s3-source}/batch/jcls"  
}
```

### iniziatori

Specifica un iniziatore batch che si avvia quando l'applicazione migrata viene avviata correttamente e continua a funzionare fino all'arresto dell'applicazione. È possibile definire una o più classi per iniziatore. È inoltre possibile definire più iniziatori. Per esempio:

```
"batch-settings": {  
  "initiators": [  
    {  
      "classes": ["A", "B"],  
      "description": "initiator...."  
    },  
    {  
      "classes": ["C", "D"],  
      "description": "initiator...."  
    }  
  ]  
}
```

```

        "description": "initiator...."
      }
    ],
    "jcl-file-location": "${s3-source}/batch/jcls"
  }

```

Per ulteriori informazioni, vedere [Per definire un iniziatore batch o una stampante SEP](#) nella documentazione di Micro Focus Enterprise Server.

- **classes**- Specifica le classi di lavoro che l'iniziatore può eseguire. È possibile utilizzare fino a 36 caratteri. È possibile utilizzare i seguenti caratteri: A-Z o 0-9.
- **description**- Descrive a cosa serve l'iniziatore.
- **jcl-file-location**- Specifica la posizione dei file JCL richiesti dai processi batch eseguiti dall'applicazione migrata.

## Impostazioni CICS: obbligatorie

Specificate i dettagli richiesti per le transazioni CICS eseguite come parte dell'applicazione utilizzando la seguente struttura.

```

"cics-settings": {
  "binary-file-location": "${s3-source}/cics/binaries",
  "csd-file-location": "${s3-source}/cics/def",
  "system-initialization-table": "BNKCICV"
}

```

### binary-file-location

Specifica la posizione dei file del programma di transazione CICS.

### csd-file-location

Specifica la posizione del file CICS Resource Definition (CSD) per questa applicazione. Per ulteriori informazioni, vedete [CICS Resource Definitions](#) nella documentazione di Micro Focus Enterprise Server.

### system-initialization-table

Specifica la tabella di inizializzazione del sistema (SIT) utilizzata dall'applicazione migrata. Il nome della tabella SIT può contenere fino a 8 caratteri. Puoi usare A-Z, 0-9, \$, @ e #. Per ulteriori



informazioni, vedete [CICS Resource Definitions](#) nella documentazione di Micro Focus Enterprise Server.

## Risorse XA: obbligatorie

Specificate i dettagli richiesti per le risorse XA richieste dall'applicazione utilizzando la seguente struttura.

```
"xa-resources" : [{
  "name": "XASQL",
  "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
  "module": "${s3-source}/xa/ESPGSQLXA64.so"
}]
```

### nome

Obbligatorio. Specifica il nome della risorsa XA.

### secret-manager-arn

Speciifica l'Amazon Resource Name (ARN) per il segreto che contiene le credenziali per la connessione al database.

### modulo

Speciifica la posizione del file eseguibile del modulo di commutazione RM. Per ulteriori informazioni, vedere [Pianificazione e progettazione di XAR](#) nella documentazione di Micro Focus Enterprise Server.

## AWSRiferimento alla definizione del set di dati per la modernizzazione del mainframe

Se la tua applicazione richiede più di pochi set di dati per l'elaborazione, inserirli uno per uno nella console di modernizzazione del AWS mainframe non è efficiente. Si consiglia invece di creare un file JSON per specificare ogni set di dati. I diversi tipi di set di dati vengono specificati in modo diverso in JSON, sebbene molti parametri siano comuni. Questo documento descrive i dettagli del JSON necessario per importare diversi tipi di set di dati.

### Note

Prima di importare qualsiasi set di dati, è necessario trasferire i set di dati dal mainframe a. AWS Quindi è necessario assicurarsi che i set di dati vengano convertiti dal formato mainframe in un formato AWS utilizzabile. Se necessario, trasforma i dati secondo necessità e archivia i set di dati trasformati in Amazon S3. Specificate il nome del bucket e della cartella nel file JSON di definizione del set di dati.

Se si utilizza il motore di runtime Micro Focus, è possibile utilizzare l'`DFCONV` utilità per convertire i set di dati. Includiamo questa utilità nelle nostre immagini Micro Focus Enterprise Developer ed Enterprise Server. Per ulteriori informazioni, vedere [DFCONV Batch File Conversion](#) nella documentazione di Micro Focus Enterprise Developer.

## Argomenti

- [Proprietà comuni](#)
- [Esempio di formato di richiesta di set di dati per VSAM](#)
- [Esempio di formato di richiesta del set di dati per GDG Base](#)
- [Esempio di formato di richiesta del set di dati per PS o GDG Generations](#)
- [Esempio di formato di richiesta del set di dati per PO](#)

## Proprietà comuni

Diversi parametri sono comuni a tutti i set di dati. Questi parametri riguardano le seguenti aree:

- Informazioni sul set di dati (`datasetName`, `datasetOrg`, `recordLength`, `encoding`)
- Informazioni sulla posizione da cui si esegue l'importazione, ovvero la posizione di origine del set di dati. Questa non è la posizione sul mainframe. È il percorso verso la posizione Amazon S3 in cui hai caricato il set di dati (`externalLocation`).
- Informazioni sulla posizione in cui stai importando, ovvero la posizione di destinazione del set di dati. Questa posizione è un database o un file system, a seconda del motore di runtime in uso. (`storageType` e `relativePath`).
- Informazioni sul tipo di set di dati (tipo specifico di set di dati, formato, codifica e così via).

Ogni definizione di set di dati ha la stessa struttura JSON. L'esempio JSON seguente mostra tutti questi parametri comuni.

```
{
  "dataSet": {
    "storageType": "Database",
    "datasetName": "MFI01V.MFIDEMO.BNKACC",
    "relativePath": "DATA",
    "datasetOrg": {
      "type": {
        type-specific properties
        ...
      },
    },
  },
}
```

Le seguenti proprietà sono comuni a tutti i set di dati.

### storageType

Obbligatorio. Si applica alla posizione di destinazione. Specifica se il set di dati è archiviato in un database o in un file system. I valori possibili sono Database o FileSystem.

- AWS Motore di runtime Blu Age: i file system non sono supportati. È necessario utilizzare un database.
- Motore di runtime Micro Focus: i database e i file system sono entrambi supportati. Puoi utilizzare Amazon Relational Database Service o Amazon Aurora per i database e Amazon Elastic File System o Amazon FSx for Lustre per i file system.

### datasetName

Obbligatorio. Specifica il nome completo del set di dati così come appare sul mainframe.

### Percorso relativo

Obbligatorio. Si applica alla posizione di destinazione. Specifica la posizione relativa del set di dati nel database o nel file system.

### DataSet.org

Obbligatorio. Specifica il tipo di set di dati. I valori possibili sono vsam, gdg, ps, po o unknown.

- AWS Motore di runtime Blu Age: sono supportati solo set di dati di tipo VSAM.
- Motore di runtime Micro Focus: sono supportati set di dati di tipo VSAM, GDG, PS, PO o Unknown.

**Note**

Se l'applicazione richiede file che non sono file di dati COBOL ma sono PDF o altri file binari, è possibile specificarli come segue:

```
"datasetOrg": {
  "type": PS {
    "format": U
  },

```

## Esempio di formato di richiesta di set di dati per VSAM

- AWSMotore di runtime Blu Age: supportato.
- Motore di runtime Micro Focus: supportato.

Se state importando set di dati VSAM, specificate `vsam` come `datasetOrg`. Il codice JSON dovrebbe essere simile al seguente esempio:

```
{
  "storageType": "Database",
  "datasetName": "AWS.M2.VSAM.KSDS",
  "relativePath": "DATA",
  "datasetOrg": {
    "vsam": {
      "encoding": "A",
      "format": "KS",
      "primaryKey": {
        "length": 11,
        "offset": 0
      }
    }
  },
  "recordLength": {
    "min": 300,
    "max": 300
  }
}
```

```
},  
"externalLocation": {  
  "s3Location": "s3://$M2_DATA_STORE/catalog/data/AWS.M2.VSAM.KSDS.DAT"  
}
```

Le seguenti proprietà sono supportate per i set di dati VSAM.

#### encoding

Obbligatorio. Specifica la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (A), EBCDIC (E) e Unknown (U).

#### format

Obbligatorio. Specifica il tipo di set di dati VSAM e il formato del record.

- AWS Motore di runtime Blu Age: i valori possibili sono ESDS (ES), KSDS (K) e RRDS (R). RR Il formato di registrazione può essere fisso o variabile.
- Motore di runtime Micro Focus: i valori possibili sono ESDS (ES), KSDS (K) e RRDS (R). RR La definizione VSAM include il formato di registrazione, quindi non è necessario specificarlo separatamente.

#### Chiave primaria

Si applica solo ai set di dati VSAM KSDS. Specifica la chiave primaria. È costituito dal nome della chiave primaria, dall'offset della chiave e dalla lunghezza della chiave. Sono name facoltativi offset e length obbligatori.

#### Lunghezza del record

Obbligatorio. Specifica la lunghezza di un record. Per i formati di record a lunghezza fissa, questi valori devono corrispondere.

- AWS Motore di runtime Blu Age: per VSAM ESDS, KSDS e RRDS, min è opzionale ed è obbligatorio. max
- Motore di runtime Micro Focus: min e max sono obbligatori.

#### Allocazione esterna

Obbligatorio. Specifica la posizione di origine, ovvero il bucket Amazon S3 in cui hai caricato il set di dati.

## Proprietà specifiche del motore Blue Age

Il motore di runtime AWS Blu Age supporta la compressione per i set di dati VSAM. L'esempio seguente mostra come specificare questa proprietà in JSON.

```
{
  common properties
  ...
  "datasetOrg": {
    "vsam": {
      common properties
      ...
      "compressed": boolean,
      common properties
      ...
    }
  }
}
```

Specificate la proprietà di compressione come segue:

compressione

Facoltativo. Specifica se gli indici per questo set di dati vengono archiviati come valori compressi. Se disponi di un set di dati di grandi dimensioni (in genere > 100 Mb), valuta la possibilità di impostare questo flag su. `true`

## Esempio di formato di richiesta del set di dati per GDG Base

- AWSMotore di runtime Blu Age: non supportato.
- Motore di runtime Micro Focus: supportato.

Se state importando set di dati di base GDG, specificate `gdg` come. `datasetOrg` Il tuo JSON dovrebbe essere simile al seguente esempio:

```
{
  "storageType": "Database",
  "datasetName": "AWS.M2.GDG",
  "relativePath": "DATA",
```

```

    "datasetOrg": {
      "gdg": {
        "limit": "3",
        "rollDisposition": "Scratch and No Empty"
      }
    }
  }
}

```

Le seguenti proprietà sono supportate per i set di dati di base GDG.

#### limit

Obbligatorio. Specifica il numero di generazioni attive, o bias. Per un cluster di base GDG, il massimo è 255.

#### Roll Disposition

Facoltativo. Specifica come gestire i set di dati di generazione quando viene raggiunto o superato il valore massimo. I valori possibili sono No Scratch and No Empty, Scratch and No Empty, Scratch and Empty o No Scratch and Empty. Il valore predefinito è Scratch and No Empty.

## Esempio di formato di richiesta del set di dati per PS o GDG Generations

- AWSMotore di runtime Blu Age: non supportato.
- Motore di runtime Micro Focus: supportato.

Se state importando set di dati di generazioni PS o GDG, specificate ps come datasetOrg Il tuo JSON dovrebbe essere simile al seguente esempio:

```

{
  "storageType": "Database",
  "datasetName": "AWS.M2.PS.FB",
  "relativePath": "DATA",
  "datasetOrg": {
    "ps": {
      "format": "FB",
      "encoding": "A"
    }
  },
}

```

```

    "recordLength": {
      "min": 300,
      "max": 300
    }
  },
  "externalLocation": {
    "s3Location": "s3://$M2_DATA_STORE/catalog/data/AWS.M2.PS.LSEQ"
  }
}

```

Le seguenti proprietà sono supportate per i set di dati di generazione PS o GDG.

#### format

Obbligatorio. Specifica il formato dei record del set di dati. I valori possibili sono FFA,FB,FBA,FBM,FBS,FM,FS,LSEQ,U,V,VA,VB,VBA,VBM, VBSVM, eVS.

#### encoding

Obbligatorio. Specifica la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (A), EBCDIC ( ) e Unknown (E) ?

#### Lunghezza del record

Obbligatorio. Specifica la lunghezza di un record. È necessario specificare la lunghezza minima (min) e massima (max) del record. Per i formati di record a lunghezza fissa, questi valori devono corrispondere.

#### Posizionamento esterno

Obbligatorio. Specifica la posizione di origine, ovvero il bucket Amazon S3 in cui hai caricato il set di dati.

## Esempio di formato di richiesta del set di dati per PO

Se state importando set di dati PO, specificate po come. datasetOrg Il tuo JSON dovrebbe essere simile al seguente esempio:

```

{
  "storageType": "Database",
  "datasetName": "AWS.M2.PO.PROC",
  "relativePath": "DATA",

```



```
"datasetOrg": {
  "po": {
    "format": "LSEQ",
    "encoding": "A",
    "memberFileExtensions": ["PRC"]
  }
},
"recordLength": {
  "min": 80,
  "max": 80
}
},
"externalLocation": {
  "s3Location": "s3://$M2_DATA_STORE/source/proc/"
}
}
```

Le seguenti proprietà sono supportate per i set di dati PO.

#### format

Obbligatorio. Specifica il formato dei record del set di dati. I valori possibili sono FFA,FB,FBA,FBM,FBS,FM,FS,LSEQ,U,V,VA,VB,VBA,VBM, VBSVM, eVS.

#### encoding

Obbligatorio. Specifica la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (A), EBCDIC (E) e Unknown (U).

#### memberFileExtensions


Obbligatorio. Specifica un array contenente una o più estensioni dei nomi di file, che consente di specificare quali file includere come membri PDS.

#### Lunghezza del record

Facoltativo. Specifica la lunghezza di un record. Sia la lunghezza minima (min) che quella massima (max) del record sono opzionali. Per i formati di record a lunghezza fissa, questi valori devono corrispondere.

#### Posizionamento esterno

Obbligatorio. Specifica la posizione di origine, ovvero il bucket Amazon S3 in cui hai caricato il set di dati.

 **Note**

L'attuale implementazione per il motore di runtime Micro Focus aggiunge le voci PDS come set di dati dinamici.

# Ambienti di runtime gestiti nella modernizzazione del AWS mainframe

Se non conosci la modernizzazione del AWS mainframe, consulta i seguenti argomenti per iniziare:

- [Cos'è la modernizzazione del AWS mainframe?](#)
- [Configurazione della modernizzazione del AWS mainframe](#)
- [Guida introduttiva alla modernizzazione del AWS mainframe](#)
- [Tutorial: Managed Runtime per AWS Blu Age](#)
- [Tutorial: runtime gestito per Micro Focus](#)

Un ambiente di runtime in AWS Mainframe Modernization è una combinazione denominata di risorse di AWS elaborazione, un motore di runtime e i dettagli di configurazione specificati dall'utente. L'ambiente di runtime ospita una o più applicazioni. Le applicazioni in AWS Mainframe Modernization contengono carichi di lavoro mainframe migrati. Puoi scegliere il motore di runtime per gli ambienti che crei. Scegliete AWS Blu Age se utilizzate il pattern di refactoring automatico e Micro Focus se utilizzate il pattern di replatforming. Puoi anche scegliere la quantità di risorse di elaborazione più adatta alla tua applicazione e, facoltativamente, collegare lo storage agli ambienti di runtime. AWS La modernizzazione del mainframe abilita i CloudWatch parametri e la registrazione di Amazon per te in modo da poter monitorare il tuo ambiente di runtime.

## Argomenti

- [Crea un ambiente di runtime per la modernizzazione del AWS mainframe](#)
- [Aggiorna un ambiente di AWS runtime per la modernizzazione del mainframe](#)
- [Arresta un ambiente di AWS runtime di modernizzazione del mainframe](#)
- [Riavviare un ambiente di AWS runtime di modernizzazione del mainframe](#)
- [Eliminare un ambiente di runtime di AWS modernizzazione del mainframe](#)

## Crea un ambiente di runtime per la modernizzazione del AWS mainframe

Utilizza la console di modernizzazione del AWS mainframe per creare un ambiente di modernizzazione del AWS mainframe.

Queste istruzioni presuppongono che tu abbia completato i passaggi indicati. [Configurazione della modernizzazione del AWS mainframe](#)

## Crea un ambiente di runtime

Per creare un ambiente di runtime

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui desiderate creare l'ambiente.
3. Nella pagina Ambienti, scegli Crea ambiente.
4. Nella pagina Specificare le informazioni di base, fornisci le seguenti informazioni:
  - a. Nella sezione Nome e descrizione, inserisci un nome per l'ambiente.
  - b. (Facoltativo) Nel campo Descrizione dell'ambiente, inserisci una descrizione per l'ambiente. Questa descrizione può aiutare te e gli altri utenti a identificare lo scopo dell'ambiente di runtime.
  - c. Nella sezione Opzioni del motore, scegliete Blu Age per il refactoring automatizzato o Micro Focus per il replatforming.
  - d. Scegliete una versione per il motore selezionato.
  - e. (Facoltativo) Nella sezione Tag, scegli Aggiungi nuovo tag per aggiungere uno o più tag di ambiente al tuo ambiente. Un tag di ambiente è un'etichetta di attributo personalizzata che consente di organizzare e gestire le AWS risorse.
  - f. Seleziona Avanti.
5. Nella pagina Specificare le configurazioni, fornisci le seguenti informazioni:
  - a. Nella sezione Disponibilità, scegli Ambiente di runtime autonomo o Cluster ad alta disponibilità.  
  
Il modello di disponibilità determina la disponibilità dell'applicazione al momento dell'esecuzione. La versione standalone va bene per scopi di sviluppo. L'alta disponibilità è per le applicazioni che devono essere sempre disponibili.
  - b. In Risorse, scegli un tipo di istanza e la capacità desiderata.


Queste risorse sono le istanze Amazon EC2 gestite dalla modernizzazione del AWS mainframe che ospiteranno il tuo ambiente di runtime. Gli ambienti di runtime autonomi

offrono due scelte: tipo di istanza e autorizzano una sola istanza. Gli ambienti di runtime ad alta disponibilità offrono due scelte per tipo di istanza e consentono fino a due istanze.

Per ulteriori informazioni, consulta la sezione [Tipi di istanze di Amazon EC2](#) e contatta uno specialista di AWS mainframe per ricevere assistenza.

6. Nella sezione Sicurezza e rete, procedi come segue:

- a. Se desideri che le applicazioni siano accessibili pubblicamente, scegli Consenti alle applicazioni distribuite in questo ambiente di essere accessibili pubblicamente.
- b. Scegli un Virtual Private Cloud (VPC).
- c. Se utilizzi il modello di alta disponibilità, scegli due o più sottoreti. Se utilizzi il pattern autonomo con il motore AWS Blu Age, scegli due o più sottoreti. Se utilizzate il pattern autonomo con il motore Micro Focus, potete specificare una sottorete.
- d. Scegli un gruppo di sicurezza per il VPC selezionato.

 Note

AWSMainframe Modernization crea un Network Load Balancer per distribuire le connessioni all'ambiente di runtime. Assicuratevi che le regole in entrata del gruppo di sicurezza consentano l'accesso da un indirizzo IP alla porta specificata nella `Listener` proprietà della definizione dell'applicazione. Per ulteriori informazioni, consulta [Registrazione degli obiettivi](#) nella Guida per l'utente di Network Load Balancers.

- e. Nel campo Chiave KMS, scegli Personalizza le impostazioni di crittografia se desideri utilizzare un servizio gestito dal cliente. AWS KMS key Per ulteriori informazioni, consulta [Crittografia dei dati inattiva per il servizio di AWS modernizzazione del mainframe](#).

 Note

Per impostazione predefinita, AWS Mainframe Modernization crittografa i tuoi dati con un file AWS KMS key che AWS Mainframe Modernization possiede e gestisce per te. Tuttavia, puoi scegliere di utilizzare un servizio gestito dal cliente. AWS KMS key

- f. (Facoltativo) Scegli un nome AWS KMS key per nome o Amazon Resource Name (ARN). In alternativa, scegli Crea un file AWS KMS key per accedere alla AWS KMS console e crearne una nuova. AWS KMS key

- g. Seleziona Avanti.
7. (Facoltativo) Nella pagina **Allega storage**, scegli uno o più file system Amazon EFS o Amazon FSx, quindi scegli **Avanti**.
  8. Nella sezione **Finestra di manutenzione**, scegli quando applicare le modifiche in sospeso all'ambiente.
    - Se scegli **Nessuna preferenza**, AWS Mainframe Modernization sceglie una finestra di manutenzione ottimizzata per te.
    - Se desideri specificare una finestra di manutenzione particolare, scegli **Seleziona nuova finestra di manutenzione**. Quindi scegli un giorno della settimana, un'ora di inizio e una durata per la finestra di manutenzione.

Per ulteriori informazioni sulla finestra di manutenzione, consulta [AWS Finestra di manutenzione della modernizzazione del mainframe](#).

Seleziona Avanti.

9. Nella pagina **Rivedi e crea**, esamina le informazioni che hai inserito, quindi scegli **Crea ambiente**.

## Aggiorna un ambiente di AWS runtime per la modernizzazione del mainframe

Utilizzate la console di modernizzazione del AWS mainframe per aggiornare un ambiente di runtime di modernizzazione AWS del mainframe. È possibile aggiornare la versione secondaria del motore di runtime o il tipo di istanza che ospita l'ambiente di runtime. È possibile scegliere se applicare gli aggiornamenti immediatamente o durante la finestra di manutenzione preferita.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

### Aggiorna un ambiente di runtime

Per aggiornare un ambiente di runtime

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).

2. Nel Regione AWS selettore, scegliete la regione in cui è stato creato l'ambiente che desiderate aggiornare.
3. Nella pagina Ambienti, scegli l'ambiente che desideri aggiornare.
4. Nella pagina dei dettagli dell'ambiente, scegli Azioni, quindi scegli Modifica ambiente.
5. Effettua una qualsiasi delle seguenti modifiche:
  - Nella sezione Opzioni del motore, scegli la versione del motore che desideri.
  - Nella sezione Risorse, scegli il tipo di istanza che desideri.
  - Nella sezione della finestra di manutenzione, scegli il giorno, l'ora e la durata che desideri.

#### Note

Le uniche modifiche che puoi scegliere di applicare durante la finestra di manutenzione sono le modifiche alla versione del motore. Tutte le altre modifiche devono essere applicate immediatamente.

6. Seleziona Avanti.
7. In Quando applicare queste modifiche, scegli Immediatamente o Durante la finestra di manutenzione successiva. Quindi scegli Aggiorna ambiente.

Se scegli Immediatamente, viene visualizzato un messaggio al termine dell'aggiornamento dell'ambiente.

## AWS Finestra di manutenzione della modernizzazione del mainframe

Ogni ambiente di runtime ha una finestra di manutenzione settimanale di un'ora. Tutte le modifiche al sistema vengono applicate durante questo periodo. La finestra di manutenzione consente di controllare quando vengono apportate modifiche e patch al software. Se un evento di manutenzione è pianificato per una determinata settimana, inizia durante quella finestra di manutenzione di un'ora. La maggior parte degli eventi di manutenzione viene completata anche durante la finestra di manutenzione di un'ora, sebbene eventi di manutenzione più grandi possano richiedere più di un'ora per essere completati.

La finestra di manutenzione di un'ora viene selezionata a caso da un periodo di 8 ore per regione. Se non si specifica una finestra di manutenzione quando si crea un ambiente di runtime, AWS

Mainframe Modernization assegna una finestra di manutenzione di 1 ora in un giorno della settimana selezionato casualmente.

AWS La modernizzazione del mainframe consuma parte delle risorse dell'istanza dell'ambiente durante la manutenzione. Durante la manutenzione si potrebbe osservare un effetto minimo sulle prestazioni o alcune interruzioni delle applicazioni.

La tabella seguente mostra i periodi di tempo predefiniti in cui vengono assegnate le finestre di manutenzione per ciascuna regione.

Nome della regione	Regione	Periodo di tempo
US East (N. Virginia)	us-east-1	03:00 - 11:00 UTC
US West (Oregon)	us-west-2	06:00 - 14:00 UTC
Asia Pacific (Mumbai)	ap-south-1	06:00 - 14:00 UTC
Asia Pacific (Singapore)	ap-southeast-1	14:00 - 22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	12:00 - 20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	13:00 - 21:00 UTC
Canada (Central)	ca-central-1	03:00 - 11:00 UTC
Europe (Frankfurt)	eu-central-1	21:00 - 05:00 UTC
Europe (Ireland)	eu-west-1	22:00 - 06:00 UTC
Europe (London)	eu-west-2	22:00 - 06:00 UTC
Europe (Paris)	eu-west-3	23:59 - 07:29 UTC
Sud America (São Paulo)	sa-east-1	00:00 - 08:00 UTC



# Arresta un ambiente di AWS runtime di modernizzazione del mainframe

Utilizzate la console di modernizzazione del AWS mainframe per interrompere un ambiente di runtime di modernizzazione AWS del mainframe. Quando si arresta un ambiente, le attuali distribuzioni delle applicazioni vengono mantenute e non verrà addebitato alcun costo per l'ambiente fino al riavvio dell'ambiente.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

## Arresta un ambiente di runtime

Se è necessario interrompere un ambiente di runtime di modernizzazione del AWS mainframe, è necessario seguire passaggi simili a quelli della sezione sull'ambiente di aggiornamento.

Utilizzate la console di modernizzazione del AWS mainframe per interrompere un ambiente di runtime di modernizzazione del AWS mainframe. Quando si arresta un ambiente, le attuali distribuzioni delle applicazioni vengono mantenute e non verrà addebitato alcun costo per l'ambiente fino al riavvio dell'ambiente.

## Arresta un ambiente di runtime

Per interrompere un ambiente di runtime di modernizzazione del AWS mainframe, seguite passaggi simili a quelli della sezione sull'ambiente di aggiornamento.

### Note

È necessario arrestare tutte le applicazioni prima di arrestare l'ambiente.

Per arrestare un ambiente di runtime

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stato creato l'ambiente che desiderate interrompere.

3. Nella pagina Ambienti, scegli l'ambiente che desideri interrompere.
4. Nella pagina dei dettagli dell'ambiente, scegli Azioni, quindi scegli Modifica ambiente.
5. Nella pagina Modifica ambiente, trova la sezione Risorse e aggiorna la capacità desiderata a zero.

#### Note

Per interrompere un ambiente, puoi solo scegliere di interromperlo immediatamente.

6. Seleziona Avanti.
7. In Quando applicare queste modifiche, scegli Immediatamente. Quindi scegli Aggiorna ambiente.

Viene visualizzato un messaggio quando la capacità dell'ambiente viene aggiornata.

## Riavviare un ambiente di AWS runtime di modernizzazione del mainframe

Utilizzate la console di modernizzazione del AWS mainframe per riavviare un ambiente di runtime di modernizzazione AWS del mainframe. Quando riavvii un ambiente di runtime, la fatturazione per l'ambiente verrà ripresa.

### Riavvia un ambiente di runtime

Per riavviare un ambiente di runtime di modernizzazione del AWS mainframe, seguite passaggi simili a quelli della sezione Stop Environment.

Per riavviare un ambiente di runtime

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/.](https://console.aws.amazon.com/m2/)
2. Nel Regione AWS selettore, scegliete la regione in cui è stato creato l'ambiente che desiderate riavviare.
3. Nella pagina Ambienti, scegli l'ambiente che desideri riavviare.
4. Nella pagina dei dettagli dell'ambiente, scegli Azioni, quindi scegli Modifica ambiente.

**Note**

La capacità desiderata per l'ambiente standalone può essere aggiornata solo a 1. Per riavviare un ambiente di runtime, puoi solo scegliere di riavviarlo immediatamente.

5. Nella pagina Modifica ambiente, trova la sezione Risorse e aggiorna la capacità desiderata da zero alla capacità richiesta.
6. Seleziona Avanti.
7. In Quando applicare queste modifiche, scegli Immediatamente. Quindi scegli Aggiorna ambiente.

Viene visualizzato un messaggio quando la capacità dell'ambiente viene aggiornata e l'ambiente viene riavviato.

## Eliminare un ambiente di runtime di AWS modernizzazione del mainframe

Utilizzate la console di modernizzazione del AWS mainframe per eliminare un ambiente di runtime di modernizzazione AWS del mainframe.

Queste istruzioni presuppongono che la procedura in [Configurazione della modernizzazione del AWS mainframe](#) sia stata completata.

### Eliminare un ambiente di runtime

Se devi eliminare un ambiente di runtime di modernizzazione del AWS mainframe, assicurati di eliminare prima tutte le applicazioni distribuite dall'ambiente. Non è possibile eliminare un ambiente di runtime in cui vengono distribuite le applicazioni.

Per eliminare un ambiente

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegliete la regione in cui è stato creato l'ambiente che desiderate eliminare.
3. Nella pagina Ambienti, scegli l'ambiente che desideri eliminare, quindi scegli Azioni ed Elimina ambiente.

4. Nella finestra Elimina ambiente, inserisci delete per confermare che desideri eliminare l'ambiente di runtime, quindi scegli Elimina.

# Test delle applicazioni nella modernizzazione del AWS mainframe

AWS Application Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

AWS Mainframe Modernization Application Testing fornisce test automatizzati di equivalenza funzionale per i progetti di migrazione.

## Argomenti

- [Che cos'è il AWS test delle applicazioni di modernizzazione del mainframe?](#)
- [AWS Concetti di test delle applicazioni di modernizzazione del mainframe](#)
- [Tutorial: configurare l'applicazione CardDemo di esempio](#)
- [Tutorial: Riproduci e confronta i test delle applicazioni di modernizzazione del AWS mainframe utilizzando CardDemo AWS Blu Age distribuito su Amazon EC2](#)
- [AWS Pagine di codici per i set di dati supportati dai test delle applicazioni di modernizzazione del mainframe](#)

## Che cos'è il AWS test delle applicazioni di modernizzazione del mainframe?

AWSApplication Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

I test influiscono in modo significativo sui progetti di migrazione. Può richiedere fino al 70% del tempo e degli sforzi del progetto di migrazione, modernizzazione o potenziamento. AWS Application Testing, una funzionalità di AWS Mainframe Modernization, fornisce test automatizzati di equivalenza funzionale per le applicazioni migrate. I test di equivalenza funzionale consentono di verificare che le applicazioni installate sul mainframe Cloud AWS siano equivalenti a quelle utilizzate sul

mainframe. AWS Application Testing confronta automaticamente le modifiche ai set di dati, ai record del database e alle schermate 3270 online tra il mainframe e AWS. Inoltre, Application Testing consente test ripetibili, in modo da poter eseguire gli scenari di test più volte man mano che si aggiorna l'architettura di destinazione, si risolvono problemi e si procede verso un'applicazione completamente migrata. Dopo la migrazione, puoi continuare a utilizzare Application Testing per i test di regressione, per assicurarti che gli aggiornamenti ai motori di runtime o ad altri componenti non causino regressioni. Il test delle applicazioni è conveniente: gli ambienti di test target vengono creati utilizzando i CloudFormation modelli forniti dall'utente, sfruttando i concetti di Infrastructure-as-Code (IaC). Application Testing accelera i progetti di migrazione utilizzando l'elasticità del cloud. È possibile eseguire scenari di test indipendenti su tutti gli ambienti paralleli necessari, riducendo le tempistiche dei test.

## Argomenti

- [Sei un utente di Application Testing per la prima volta?](#)
- [Vantaggi del test delle applicazioni](#)
- [Integrazione con AWS CloudFormation](#)
- [Come funziona il test delle applicazioni](#)
- [Servizi correlati](#)
- [Accesso ai test delle applicazioni](#)
- [Prezzi per il test delle applicazioni](#)

## Sei un utente di Application Testing per la prima volta?

Se sei un utente alle prime armi di Application Testing, ti consigliamo di iniziare leggendo le seguenti sezioni:

- [Concetti di test delle applicazioni](#)
- [Tutorial: Configurazione CardDemo](#)

## Vantaggi del test delle applicazioni

Application Testing offre diversi vantaggi per aiutarvi nel processo di migrazione:

- Test di accelerazione, agilità e flessibilità
- Concetti di test «Registra una volta sul mainframe, riproduci più volte in AWS»

- Creazione IaC di ambienti target tramite modelli forniti dall'utente CloudFormation
- Elevati gradi di ripetibilità dei test
- Progettato per il cloud, con scalabilità ed elasticità in mente
- Test su larga scala con elevato grado di automazione
- Efficienza dei costi

## Integrazione con AWS CloudFormation

Application Testing utilizza l'infrastruttura come codice con AWS CloudFormation. Questa scelta progettuale semplifica e migliora l'esperienza di test. AWS CloudFormation ti offre l'autonomia e l'indipendenza necessarie per definire l'infrastruttura migliore per le tue esigenze. È possibile selezionare o definire diversi parametri (dimensione dell'istanza, istanza RDS, gruppo di sicurezza ottimale) in modo indipendente. Puoi aggiungere risorse, ad esempio una coda Amazon SQS, di cui hai bisogno per il corretto funzionamento dell'applicazione in condizioni di test.

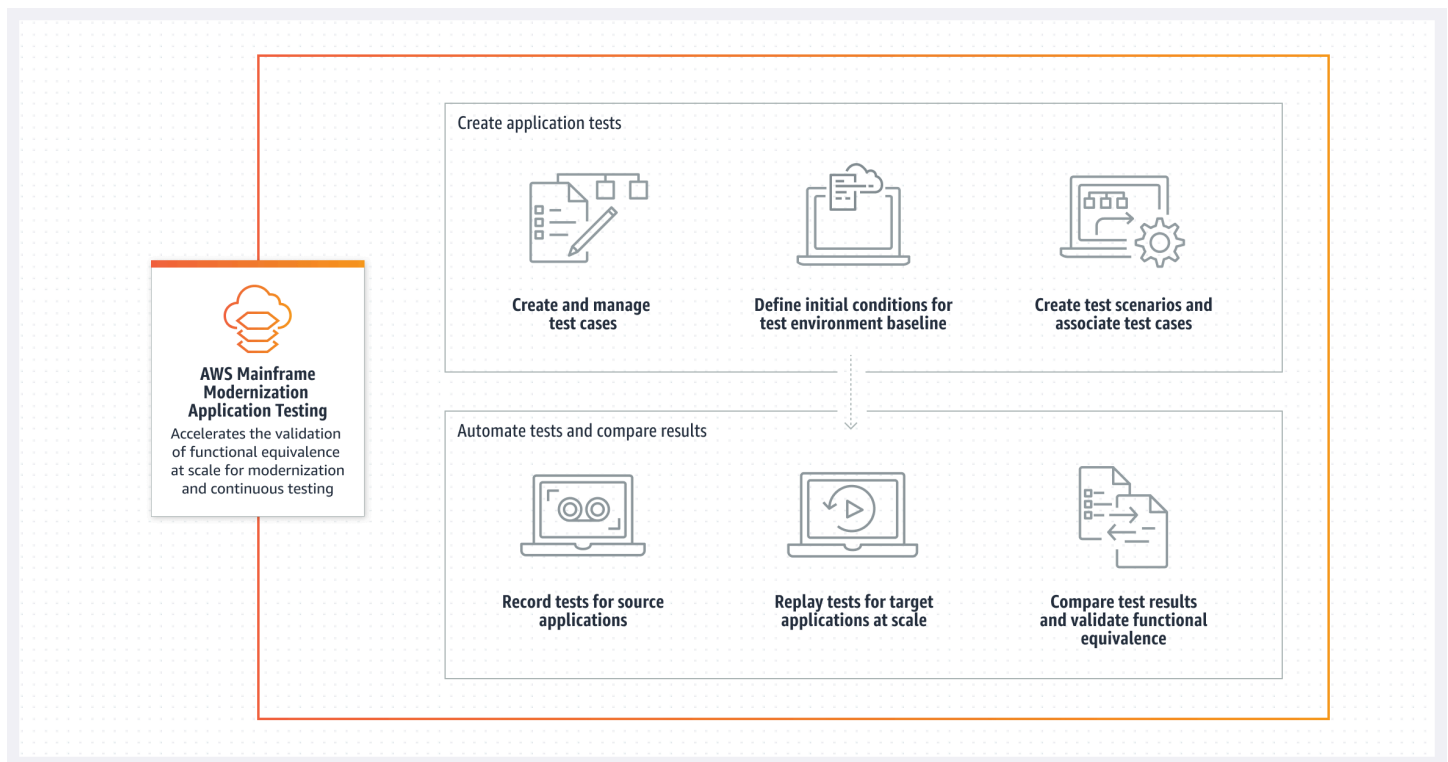
Nei AWS CloudFormation modelli disponibili per il download, noterai alcune caratteristiche comuni:

- Application Testing crea uno stack completamente isolato, che include un ambiente di runtime e un'applicazione di modernizzazione del AWS mainframe, con definizioni di rete e sicurezza proprie. Questo stack isolato fornisce resilienza, perché gli altri attori dello stesso Account AWS non possono interferire con l'attività di test. Evita inoltre situazioni in cui gli operatori di sistema modificano il VPC o il gruppo di sicurezza predefinito, il che può causare errori nell'attività di test.
- Il gruppo di sicurezza consente inoltre di controllare l'accesso esterno alle risorse utilizzate nei test. Ad esempio, un database potrebbe contenere dati riservati.
- L'isolamento completo impedisce ad altri attori che condividono il VPC di spiare il traffico.
- Migliora le prestazioni. Ad esempio, la comunicazione tra l'applicazione AWS Mainframe Modernization creata dal modello e il relativo database Amazon RDS avviene su una rete separata (un VPC privato), che evita che altri attori rallentino il traffico.

Ti consigliamo di implementare queste funzionalità anche nei modelli che crei AWS CloudFormation.

## Come funziona il test delle applicazioni

La figura seguente è una panoramica di come funzionano i test di equivalenza funzionale in Application Testing.



- - È possibile trasferire i dati di input dalla fonte all'AWS utilizzando di AWS Mainframe Modernization [File Transfer](#) o degli strumenti preferiti per il trasferimento dei dati mainframe.
- La stessa logica aziendale viene applicata sia all'origine che alla destinazione.
- Application Testing confronta automaticamente i dati di output (set di dati, modifiche al database relazionale, schermate 3270 e interazioni utente) sia dall'origine che dalla destinazione. Dopo aver eseguito lo scenario di test sul mainframe, acquisite i dati di output e li trasferite su AWS, quindi riproducete lo scenario di test sulla destinazione. Application Testing confronta automaticamente i dati di output del test in corso AWS con i dati di output della fonte. Puoi vedere a colpo d'occhio quali record sono identici, equivalenti, diversi o mancanti. Inoltre, è possibile definire regole di equivalenza, in modo che i record che non sono identici ma hanno lo stesso significato commerciale siano considerati equivalenti.

Il flusso di lavoro che segue in Application Testing prevede i seguenti passaggi:

1. Crea casi di test. I casi di test sono l'unità più piccola di azioni di test. Quando si crea un test case, si identificano anche i tipi di dati da confrontare che meglio rappresentano l'equivalenza funzionale tra l'origine e la destinazione.
2. Crea scenari di test. Gli scenari di test raggruppano i casi di test correlati in una sequenza specifica per l'esecuzione.



3. Crea condizioni iniziali. Le condizioni iniziali spiegano come registrare un test sul mainframe e AWS CloudFormation utilizzarlo per creare automaticamente lo stesso stato suAWS.
4. Registra sulla sorgente e riproduci sulla destinazione. Acquisisci i set di dati di input e output sul mainframe e caricali su. AWS Quindi riattiva lo scenario di test. AWS
5. Confronta i set di dati di origine e di destinazione. Application Testing confronta automaticamente i set di dati di output sia dall'origine che dalla destinazione, in modo da poter vedere a colpo d'occhio cosa è corretto e cosa no.

Sia l'azione finale di uno scenario di test che l'obiettivo dell'intero processo consiste nell'identificare le discrepanze tra le esecuzioni del test di origine e quelle di destinazione. Application Testing confronta la versione di origine e la versione di destinazione per i dati acquisiti su tutti i canali di interazione durante l'esecuzione del test. Inoltre confronta gli stati finali dei dati rilevanti (come definiti nei casi di test).

## Servizi correlati

Il test delle applicazioni è una funzionalità di AWS Mainframe Modernization. Utilizza inoltre l'infrastruttura come codice AWS CloudFormation per garantire la ripetibilità dei test, l'automazione e l'efficienza dei costi. Per ulteriori informazioni, consultare:

- [Modernizzazione del mainframe AWS](#)
- [AWS CloudFormation](#)

## Accesso ai test delle applicazioni

È possibile accedere ai test delle applicazioni dalla console di modernizzazione del AWS mainframe selezionando Application Testing nella barra di navigazione a sinistra.

## Prezzi per il test delle applicazioni

I prezzi per i test delle applicazioni sono disponibili nella pagina [AWS Mainframe Modernization Pricing](#).

# AWS Concetti di test delle applicazioni di modernizzazione del mainframe

AWS Application Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

AWS Application Testing utilizza termini che altri servizi di test o pacchetti software potrebbero utilizzare con un significato leggermente diverso. Le sezioni seguenti spiegano come AWS Mainframe Modernization Application Testing utilizza questa terminologia.

## Argomenti

- [Caso di test](#)
- [Scenario di test](#)
- [Progetto di test](#)
- [Condizione iniziale](#)
- [Registrazione \(acquisizione\)](#)
- [Riproduci di nuovo](#)
- [Confronta](#)
- [Confronti tra database](#)
- [Confronti tra set di dati](#)
- [Stato del confronto](#)
- [Regole di equivalenza](#)
- [Confronto tra set di dati dello stato finale](#)
- [Confronti tra database State-Progress](#)
- [Equivalenza funzionale \(FE\)](#)
- [Confronti tra schermate 3270 online](#)
- [Registrazione](#)
- [Riproduci i dati](#)
- [Dati di riferimento](#)
- [Registra, riproduci e confronta](#)

- [Differenze](#)
- [Equivalenze](#)
- [Applicazione di origine](#)
- [Applicazione di destinazione](#)

## Caso di test

Un test case è l'unità d'azione individuale più importante nel flusso di lavoro di test. Di solito, un test case viene utilizzato per rappresentare un'unità indipendente di logica aziendale che modifica i dati. Verranno effettuati confronti per ogni test case. I casi di test vengono aggiunti a uno scenario di test. I test case contengono metadati sugli artefatti di dati (set di dati, database) modificati dal test case e sulle funzioni aziendali che vengono attivate durante l'esecuzione del test case: processi in batch, 3270 finestre di dialogo interattive e altro. Ad esempio, i nomi e le code page dei set di dati.

Dati di input → Test case → Dati di output

I casi di test possono essere online o di tipo batch:

- I test case online sono casi di test in cui l'utente esegue schermate interattive (3270) per leggere, modificare o produrre nuovi dati aziendali (record di database e/o set di dati).
- I batch test case sono casi di test che richiedono l'invio di un batch per leggere, elaborare e modificare o produrre nuovi dati aziendali (set di dati e/o record di database).

## Scenario di test

Gli scenari di test sono una serie di casi di test eseguiti in ordine sequenziale, uno per uno. La riproduzione viene eseguita a livello di scenario di test. Tutti i casi di test nello scenario di test vengono eseguiti nell'ambiente di test di destinazione quando viene ripetuto uno scenario di test. Se ci sono differenze dopo aver confrontato gli artefatti dei test di riferimento e di ripetizione, le differenze verranno mostrate a livello di test case.

Ad esempio, Scenario di test A:

Test Case 1, Test Case 2, Test Case 3 e così via.

## Progetto di test

I progetti di test rappresentano una raccolta di scenari di test per raggiungere il traguardo di test desiderato. Ad esempio, la migrazione di un'applicazione specifica può essere considerata come un singolo progetto di test. Il raggruppamento degli scenari di test in progetti di test consente ai responsabili dei test di tenere traccia dello stato del progetto di test, compresi i test superati/non riusciti.

## Condizione iniziale

Una condizione iniziale contiene un set di risorse (elaborazione, datastore e altre) da creare e dati dell'applicazione che è necessario ripristinare su tali risorse create prima di poter eseguire scenari di test. Questo crea la base di riferimento dell'ambiente di test di destinazione. Ti consente di fornire un AWS CloudFormation modello. Utilizzate il modello per creare l'ambiente di test di destinazione e, facoltativamente, l'estrazione DDL dal database di origine se lo scenario di test modifica i record del database. Ogni scenario di test verrà associato a una condizione iniziale. Una condizione iniziale può essere associata a più scenari di test. Per garantire la ripetibilità e la coerenza dei risultati ed evitare falsi positivi dovuti a dati già modificati, è necessario ripristinare le condizioni iniziali prima dell'esecuzione di ogni scenario di test.

Per gli scenari di test che contengono casi di test che modificano i record del database, la condizione iniziale fa riferimento anche a un'esportazione DDL degli schemi e delle tabelle del database di origine.

## Registrazione (acquisizione)

Le registrazioni vengono eseguite a livello di scenario di test. Durante la registrazione, devi fornire una posizione Amazon S3 che contenga gli artefatti, i set di dati e le riviste CDC del database relazionale dal mainframe di origine con cui confrontare. Questi verranno considerati dati di riferimento provenienti dal mainframe di origine. Durante la riproduzione, i dati di replay generati verranno confrontati con i dati di riferimento registrati per garantire l'equivalenza dell'applicazione.

## Riproduci di nuovo

I replay vengono eseguiti a livello di scenario di test. Durante la riproduzione, AWS Mainframe Modernization Application Testing utilizza lo CloudFormation script a cui si fa riferimento nella condizione iniziale associata per creare l'ambiente di test di destinazione ed eseguire l'applicazione. I set di dati e i record del database modificati durante la riproduzione vengono acquisiti e confrontati

con i dati di riferimento del mainframe. In genere, si registra sul mainframe una sola volta e poi si esegue la riproduzione più volte, fino al raggiungimento dell'equivalenza funzionale.

## Confronta

I confronti vengono effettuati automaticamente al termine di un replay. Durante i confronti, i dati di riferimento caricati e acquisiti durante la fase di registrazione vengono confrontati con i dati di replay generati durante la fase di replay. I confronti avvengono separatamente a livello di singolo test case per set di dati, record di database e schermate online.

## Confronti tra database

Application Testing utilizza una funzionalità di confronto dello stato di avanzamento quando si confrontano le modifiche nei record del database tra le applicazioni di origine e di destinazione. La corrispondenza dello stato di avanzamento confronta le differenze in ogni singola istruzione Run INSERT, UPDATE e DELETE, a differenza del confronto delle righe della tabella alla fine del processo. La corrispondenza tra stati e progressi è più efficiente rispetto alle alternative e fornisce confronti più rapidi e accurati confrontando solo i dati modificati e rilevando gli errori di correzione automatica nel flusso delle transazioni. Utilizzando la tecnologia CDC (Changed Data Capture), Application Testing è in grado di rilevare le singole modifiche del database di relazioni e confrontarle tra l'origine e la destinazione.

Le modifiche al database di relazione vengono generate sull'origine e sulla destinazione dal codice dell'applicazione testato utilizzando istruzioni DML (Data Modification Language) come SQL INSERT, UPDATE o DELETE, ma anche indirettamente quando l'applicazione utilizza procedure memorizzate o quando i trigger del database sono impostati su alcune tabelle o quando si utilizza CASCADE DELETE per garantire l'integrità referenziale, attivando automaticamente eliminazioni aggiuntive.

## Confronti tra set di dati

Application Testing confronta automaticamente i set di dati di riferimento e di riproduzione prodotti sui sistemi di origine (registrazione) e di destinazione (replay).

Per confrontare i set di dati:

1. Inizia con gli stessi dati di input (set di dati, database) sia sull'origine che sulla destinazione.
2. Esegui i tuoi casi di test sul sistema di origine (mainframe).
3. Acquisisci i set di dati prodotti e caricali in un bucket Amazon S3. Puoi trasferire i set di dati di input dalla fonte all' AWS utilizzo di riviste, schermate e set di dati CDC.

4. Specificate la posizione del bucket Amazon S3 in cui sono stati caricati i set di dati del mainframe al momento della registrazione del test case.

Una volta completata la riproduzione, Application Testing confronta automaticamente i set di dati di riferimento e di destinazione in uscita, mostrando se i record sono identici, equivalenti, diversi o mancanti. Ad esempio, i campi data relativi al momento dell'esecuzione del carico di lavoro (giorno+1, fine del mese corrente, ecc.) vengono automaticamente considerati equivalenti. Inoltre, è possibile definire facoltativamente regole di equivalenza, in modo che i record non identici abbiano lo stesso significato commerciale e vengano contrassegnati come equivalenti.

## Stato del confronto

Application Testing utilizza i seguenti stati di confronto: IDENTICO, EQUIVALENTE e DIVERSO.

### IDENTICO

I dati di origine e di destinazione sono esattamente gli stessi.

### EQUIVALENTE

I dati di origine e di destinazione contengono false differenze considerate equivalenti, ad esempio date o timestamp che non influiscono sull'equivalenza funzionale quando sono relative al momento dell'esecuzione del carico di lavoro. È possibile definire regole di equivalenza per identificare quali sono queste differenze. Quando tutti gli scenari di test ripetuti rispetto ai relativi scenari di test di riferimento mostrano lo stato IDENTICO o EQUIVALENTE, lo scenario di test dimostra l'equivalenza funzionale.

### DIFFERENTE

I dati di origine e di destinazione contengono differenze, ad esempio un numero diverso di record in un set di dati o valori diversi nello stesso record.

## Regole di equivalenza

Un insieme di regole per identificare le false differenze che possono essere considerate risultati equivalenti. Il test di equivalenza funzionale offline (OFET) causa inevitabilmente differenze per alcuni risultati tra il sistema di origine e quello di destinazione. Ad esempio, i timestamp di aggiornamento sono diversi in base alla progettazione. Le regole di equivalenza spiegano come correggere tali differenze ed evitare falsi positivi al momento del confronto. Ad esempio, se una data è di runtime +

2 giorni in una particolare colonna di dati, la regola di equivalenza la descrive e accetta un orario sul sistema di destinazione che corrisponde a runtime su target + 2 giorni anziché un valore strettamente uguale alla stessa colonna nella registrazione di riferimento.

## Confronto tra set di dati dello stato finale

Lo stato finale dei set di dati che sono stati creati o modificati, incluse tutte le modifiche o gli aggiornamenti apportati ai set di dati rispetto al loro stato iniziale. Per quanto riguarda i set di dati, Application Testing esamina i record in tali set di dati al termine di un test case e confronta i risultati.

## Confronti tra database State-Progress

Confronti delle modifiche apportate ai record del database come sequenza di singole istruzioni DML (Delete, Update, Insert). Application Testing confronta le singole modifiche (inserimento, aggiornamento o eliminazione della riga di una tabella) dal database di origine al database di destinazione e identificherà le differenze per ogni singola modifica. Ad esempio, è possibile utilizzare una singola istruzione INSERT per inserire in una tabella una riga con valori diversi nel database di origine rispetto al database di destinazione.

## Equivalenza funzionale (FE)

Due sistemi sono considerati funzionalmente equivalenti se producono gli stessi risultati su tutte le operazioni osservabili, con gli stessi dati di input. Ad esempio, due applicazioni sono considerate funzionalmente equivalenti se gli stessi dati di input producono dati di output identici (tramite schermate, modifiche al set di dati o modifiche al database).

## Confronti tra schermate 3270 online

Confronta l'output delle schermate mainframe 3270 con l'output delle schermate Web delle applicazioni modernizzate quando il sistema di destinazione è in esecuzione con il runtime Blu Age in AWS . Cloud AWS Inoltre, confronta l'output degli schermi mainframe 3270 con gli schermi 3270 dell'applicazione riospitata quando il sistema di destinazione è in esecuzione con il runtime Micro Focus in. Cloud AWS

## Registrazione

L'azione di ripristinare uno stato noto dei dati, quindi acquisire o registrare i dati di riferimento di uno scenario di test di riferimento (per uno o più casi di test in sequenza) su un sistema di origine.

## Riproduci i dati

I dati di riproduzione vengono utilizzati per descrivere i dati generati dalla riproduzione di uno scenario di test nell'ambiente di test di destinazione. Ad esempio, i dati di replay vengono generati quando uno scenario di test è in esecuzione su un'applicazione del servizio di modernizzazione del AWS mainframe. I dati di replay vengono quindi confrontati con i dati di riferimento acquisiti dalla fonte. Ogni volta che riproduci il carico di lavoro nell'ambiente di destinazione, viene generata una nuova generazione di dati di replay.

## Dati di riferimento

I dati di riferimento vengono utilizzati per descrivere i dati acquisiti sul mainframe di origine. È il riferimento con cui verranno confrontati i dati generati dal replay (target). Di solito, per ogni record sul mainframe che crea dati di riferimento, ci saranno molti replay. Questo perché gli utenti in genere acquisiscono lo stato corretto dell'applicazione sul mainframe e riproducono i casi di test sull'applicazione modernizzata di destinazione per convalidare l'equivalenza. Se vengono rilevati bug, vengono corretti e i test case vengono ripetuti nuovamente. Spesso, si ripetono più cicli di riproduzione, si correggono i bug e si ripete nuovamente per convalidare l'occorrenza. Si tratta del cosiddetto paradigma di test «capture once, replay multiple times».

## Registra, riproduci e confronta

Il test delle applicazioni si svolge in tre fasi:

- Record: acquisisce i dati di riferimento creati sul mainframe per ogni test case di uno scenario di test. Questi possono includere 3270 schermate online, set di dati e record di database.
  - Per le schermate 3270 online, è necessario utilizzare l'emulatore di terminale Blu Insights per acquisire il carico di lavoro di origine. Per ulteriori informazioni, consulta la documentazione di [Blu Insights](#).
  - Per i set di dati, sarà necessario acquisire i set di dati prodotti da ciascun test case sul mainframe utilizzando strumenti comuni, come FTP o il servizio di trasferimento dei set di dati, parte di AWS Mainframe Modernization.
  - Per le modifiche al database, utilizza la documentazione [AWS Mainframe Modernization Data Replication with Precisly](#) per acquisire e generare diari CDC contenenti le modifiche.
- Replay: lo scenario di test viene riprodotto nell'ambiente di destinazione. Vengono eseguiti tutti i casi di test specificati nello scenario di test. I tipi di dati specificati creati dai singoli casi di test, come set di dati, modifiche al database relazionale o schermate 3270, verranno acquisiti



con l'automazione. Questi dati sono noti come dati di replay e verranno confrontati con i dati di riferimento acquisiti durante la fase di registrazione.

#### Note

Le modifiche al database relazionale richiederanno opzioni di configurazione specifiche per DMS nel modello di condizione iniziale. CloudFormation

- **Confronta:** i dati di riferimento del test di origine e i dati di replay di destinazione vengono confrontati e i risultati verranno visualizzati come dati identici, diversi, equivalenti o mancanti.

## Differenze

Indica che sono state rilevate differenze tra i set di dati di riferimento e quelli di riproduzione mediante il confronto dei dati. Ad esempio, un campo in una schermata 3270 online che mostra valori diversi dal punto di vista della logica aziendale tra il mainframe di origine e l'applicazione modernizzata di destinazione verrà considerato una differenza. Un altro esempio è un record in un set di dati che non è identico tra le applicazioni di origine e di destinazione.

## Equivalenze

I record equivalenti sono record diversi tra i set di dati di riferimento e di riproduzione, ma non devono essere considerati diversi dal punto di vista della logica aziendale. Ad esempio, un record contenente il timestamp di quando è stato prodotto il set di dati (tempo di esecuzione del carico di lavoro). Utilizzando regole di equivalenza personalizzabili, è possibile indicare ad Application Testing di trattare tale differenza di falsi positivi come un'equivalenza, anche se mostra valori diversi tra i dati di riferimento e quelli di riproduzione.

## Applicazione di origine

L'applicazione mainframe di origine con cui confrontare.

## Applicazione di destinazione

L'applicazione nuova o modificata su cui viene eseguito il test e che verrà confrontata con l'applicazione di origine per rilevare eventuali difetti e ottenere l'equivalenza funzionale tra le applicazioni di origine e di destinazione. L'applicazione di destinazione è in genere in esecuzione nel AWS cloud.

# Tutorial: configurare l'applicazione CardDemo di esempio

AWSApplication Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

Per questo tutorial, creerete uno AWS CloudFormation stack che vi aiuti a configurare l'[applicazione di CardDemo esempio](#) per la ripiattaforma con il servizio gestito Micro Focus on AWS Mainframe Modernization e funzionalità tra cui AWS il test delle applicazioni di modernizzazione del mainframe. Il tutorial descrive un AWS CloudFormation modello di esempio che è possibile utilizzare per creare lo stack. Forniamo anche un file compresso degli artefatti applicativi necessari. Il modello di esempio fornisce un database, un ambiente di runtime, un'applicazione e un ambiente di rete completamente isolato.

Questo modello crea diverse AWS risorse. Ti verranno fatturate se crei uno stack partendo da questo modello.

## Prerequisiti

- Scarica e decomprimi il file e. [IC3-card-demo-zip\\_datasets\\_Mainframe\\_ebcdic.zip](#) Questi file contengono il CardDemo campione e i set di dati di esempio da utilizzare con AWS Application Testing.
- Crea un bucket Amazon S3 per contenere i CardDemo file e altri elementi. Ad esempio, my-carddemo-bucket.

## Fase 1: Prepararsi alla configurazione CardDemo

Carica i file di CardDemo esempio e modifica il AWS CloudFormation modello che creerà l' CardDemoapplicazione.

1. Carica le IC3-card-demo cartelle datasets\_Mainframe\_ebcdic e le cartelle che hai decompresso in precedenza nel tuo bucket.
2. Scarica il aws-m2-math-mf-carddemo.yaml AWS CloudFormation modello dal tuo bucket. È nella IC3-card-demo cartella.
3. Modifica il aws-m2-math-mf-carddemo.yaml AWS CloudFormation modello come segue:

- Cambia il `BucketName` parametro con il nome del bucket che hai definito in precedenza, ad esempio `my-carddemo-bucket`.
- Imposta `ImportJsonPath` la posizione nel bucket del `mf-carddemo-datasets-import.json` file. Ad esempio, `s3://my-carddemo-bucket/IC3-card-demo/mf-carddemo-datasets-import.json` l'aggiornamento di questo valore assicura che l'output `M2ImportJson` abbia il valore corretto.
- (Facoltativo) Adattate i `InstanceType` parametri `EngineVersion` and in modo che corrispondano ai vostri standard.

#### Note

Non modificate le `M2ApplicationId` uscite `M2EnvironmentId` e. `Application Testing` utilizza questi valori per individuare le risorse con cui interagirà.

## Fase 2: Creare tutte le risorse necessarie

Esegui il tuo AWS CloudFormation modello personalizzato per creare tutte le risorse necessarie per completare con successo questo tutorial. Questo modello configura l' `CardDemo` applicazione in modo da poterla utilizzare durante i test.

1. Accedi alla AWS CloudFormation console e scegli `Crea stack`, quindi scegli `Con nuove risorse (standard)`.
2. In `Prerequisito - Prepara il modello`, scegli `Il modello è pronto`.
3. In `Specificare il modello`, scegli `Carica un file modello`, quindi scegli `Scegli file`.
4. Vai al punto in cui lo hai scaricato `aws-m2-math-mf-carddemo.yaml` e scegli il file, quindi scegli `Avanti`.
5. In `Specificare i dettagli dello stack`, inserisci un nome per lo stack in modo da poterlo trovare facilmente in un elenco, quindi scegli `Avanti`.
6. In `Configura le opzioni dello stack`, mantieni i valori predefiniti e scegli `Avanti`.
7. In `Review`, controlla AWS CloudFormation cosa stai creando per te, quindi scegli `Invia`.

La creazione dello stack richiede circa 10-15 AWS CloudFormation minuti.

**Note**

Il modello è configurato per aggiungere un suffisso univoco ai nomi delle risorse che crea. Ciò significa che è possibile creare più istanze di questo modello di stack in parallelo, una funzionalità chiave per il test delle applicazioni che consente di eseguire più scenari di test contemporaneamente.

## Fase 3: Implementazione e avvio dell'applicazione

Distribuisci l' CardDemo applicazione che hai AWS CloudFormation creato per te e assicurati che sia in esecuzione.

1. Apri la console di modernizzazione del AWS mainframe e scegli Applicazioni dalla barra di navigazione a sinistra.
2. Scegli l' CardDemo applicazione, che ha un nome simile. `aws-m2-math-mf-carddemo-abc1d2e3`
3. Scegli Azioni, quindi scegli Distribuisci applicazione.
4. In Ambienti, scegli l'ambiente di runtime che corrisponde all'applicazione. Avrà lo stesso identificatore univoco aggiunto alla fine del nome. Ad esempio, `aws-m2-math-mf-carddemo-abc1d2e3`.
5. Seleziona Deploy (Implementa). Attendi che l'applicazione venga distribuita correttamente e che sia nello stato Pronto.
6. Scegli l'applicazione, quindi scegli Azioni e Avvia applicazione. Attendi che l'applicazione sia nello stato In esecuzione.
7. Nella pagina dei dettagli dell'applicazione, copia la porta e il nome host DNS, necessari per connetterti all'applicazione in esecuzione.

## Fase 4: Importazione dei dati iniziali

Per utilizzare l'applicazione CardDemo di esempio, è necessario importare un set iniziale di dati. Completa questa procedura:

1. Scarica il file `mf-carddemo-datasets-import.json`.
2. Modifica il file nel tuo editor di testo preferito.

3. Individua il `s3Location` parametro e aggiorna il valore in modo che punti al bucket Amazon S3 che hai creato.
4. Apporta la stessa modifica per tutte le occorrenze di `dis3Location`, quindi salva il file.
5. Accedi alla console Amazon S3 e vai al bucket creato in precedenza.
6. Carica il file personalizzato `mf-carddemo-datasets-import.json`.
7. Apri la console di modernizzazione del AWS mainframe e scegli Applicazioni dalla barra di navigazione a sinistra.
8. Scegli l'applicazione. CardDemo
9. Scegli Set di dati, quindi scegli Importa.
10. Vai alla posizione in Amazon S3 in cui hai caricato il file JSON personalizzato e scegli Invia.

Questo lavoro importa 23 set di dati. Per monitorare l'esito del processo di importazione, controlla la console. Quando tutti i set di dati sono stati importati correttamente, connettiti all'applicazione.

#### Note

Quando utilizzate questo modello in Application Testing, Output gestisce `M2ImportJson` automaticamente il processo di importazione.

## Fase 5: Connect all' CardDemoapplicazione

Connect all'applicazione CardDemo di esempio utilizzando l'emulatore 3270 di vostra scelta.

- Quando l'applicazione è in esecuzione, utilizzate l'emulatore 3270 per connettervi all'applicazione, specificando il nome host DNS e il nome della porta, se necessario.

Ad esempio, se si utilizza l'[emulatore c3270](#) open source, il comando è simile al seguente:

```
c3270 -port port-number DNS-hostname
```

port

La porta specificata nella pagina dei dettagli dell'applicazione. Ad esempio, 6000.

## Hostname (Nome host)

Il nome host DNS specificato nella pagina dei dettagli dell'applicazione.

La figura seguente mostra dove trovare la porta e il nome host DSN.

The screenshot displays the AWS Mainframe Modernization console interface. At the top, the breadcrumb navigation shows 'AWS Mainframe Modernization > Applications > aws-m2-math-mf-carddemo-7f28a650'. The main header shows the application name 'aws-m2-math-mf-carddemo-7f28a650' with an 'Info' link and an 'Actions' dropdown menu. Below the header, there are tabs for 'Definition', 'Batch jobs', 'Data sets', and 'Tags'. The 'Definition' tab is active, showing 'Application information' with an 'Info' link. The application details are organized into a grid:

<b>Name</b> aws-m2-math-mf-carddemo-7f28a650	<b>Status</b> Running	<b>Ports</b> 7000	<b>Logs</b> <a href="#">ConsoleLog</a> <a href="#">BatchJobLogs</a>
<b>ARN</b> arn:aws:m2:us-west-2:██████████:app/efzlb7ocfb5zi7fwfcxvfusw4	<b>Creation time</b> May 2, 2023 at 10:50 (UTC-04:00)	<b>KMS key</b> AWS owned key	<b>Description</b> m2 application: aws-m2-math-mf-carddemo-7f28a650
<b>Engine</b> Micro Focus	<b>DNS Hostname</b> haytgmjvgazteoi-ibgcq4di.m2.us-west-2.amazonaws.com		

## Tutorial: Riproduci e confronta i test delle applicazioni di modernizzazione del AWS mainframe utilizzando CardDemo AWS Blu Age distribuito su Amazon EC2

AWSApplication Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

In questo tutorial, completerai i passaggi necessari per riprodurre e confrontare i carichi di lavoro di test con l' CardDemo applicazione in esecuzione su AWS Blu Age distribuito su Amazon EC2.

## Fase 1: Ottenere una Amazon Machine Image (AMI) di Amazon EC2 AWS Blu Age

Segui le istruzioni nel tutorial di [installazione di AWSAWS Blu Age Runtime \(su Amazon EC2\)](#) per i passaggi di onboarding necessari per accedere a AWS Blu Age sull'AMI Amazon EC2.

## Fase 2: Avvia un'istanza Amazon EC2 utilizzando l'AMI AWS Blu Age

1. Configura le tue AWS credenziali.
2. Identifica la posizione del file binario dell'AMI Amazon EC2 3.5.0 (solo CLI/versione Blu AWS Age) dal bucket Amazon S3:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/  
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/3.5.0/AMI/
```

### Note

La funzionalità Application Testing può essere utilizzata solo in 4 regioni in prod (us-east-1, sa-east-1, eu-central-1 e ap-southeast-2).

3. Ripristina l'AMI nel tuo account con il seguente comando:

```
aws ec2 create-restore-image-task --object-key 3.5.0/AMI/ami-0182ffe3b9d63925b.bin  
--bucket aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1 --region eu-west-1 --name  
"AWS BLUAGE RUNTIME AMI"
```

### Note

Sostituisci il nome del file bin AMI e la regione in cui desideri creare l'AMI.


4. Dopo aver creato un'istanza Amazon EC2, puoi trovare l'ID AMI corretto che è stato ripristinato dall'AMI dal bucket Amazon S3 nel catalogo di immagini di Amazon EC2.

### Note

In questo tutorial, l'ID AMI è ami-0d0fafcc636fd1e6d e devi modificare questo ID nei diversi file di configurazione rispetto a quello fornito.

1. Se `aws ec2 create-restore-image-task` fallisce, controlla la tua versione di Python e CLI usando il seguente comando:

```
aws --version
```

 Note

La versione Python deve essere  $\geq 3$  e la versione CLI deve essere  $\geq 2$ .

2. Se queste versioni sono obsolete, la CLI deve essere aggiornata. Per aggiornare la CLI:
  - a. Segui le istruzioni in [Installa o aggiorna l'ultima versione dell'interfaccia a riga di comando di AWS](#).
  - b. Rimuovi CLI v1 con il seguente comando:

```
sudo yum remove awscli
```

- c. E installa CLI v2 con i seguenti comandi:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

- d. Infine, controlla la versione di Python e CLI con il seguente comando:

```
aws --version
```

3. È quindi possibile rifare `aws ec2. create-restore-image-task`

## Passaggio 3: carica i file CardDemo dipendenti su S3

Copia il contenuto di cartelle, database, file system e dati utente. Scarica e decomprimi le applicazioni. CardDemo Queste tre cartelle devono essere copiate in uno dei bucket denominato `your-s3-bucket` in questa documentazione.



## Fase 4: Caricare i database e inizializzare l'applicazione CardDemo

Crea un'istanza Amazon EC2 temporanea da utilizzare come risorsa di calcolo per generare gli snapshot del database richiesti per l'applicazione. CardDemo Questa istanza EC2 non eseguirà l' CardDemo applicazione stessa, ma genererà gli snapshot del database che verranno utilizzati in seguito.

Inizia modificando il CloudFormation modello fornito denominato 'load-and-create-ba-snapshots.yml'. Questo è il CloudFormation modello utilizzato per creare l'istanza Amazon EC2 utilizzata per generare gli snapshot del database.

1. Genera e fornisci la tua coppia di chiavi EC2 che verrà utilizzata per l'istanza EC2. Per ulteriori informazioni, consulta [Creare coppie di chiavi](#).

Esempio:

```
Ec2KeyPair:
  Description: 'ec2 key pair'
  Default: 'm2-tests-us-west-2'
  Type: String
```

2. Specificate il percorso Amazon S3 della cartella in cui avete inserito la cartella del database nel passaggio precedente:

```
S3DBScriptsPath:
  Description: 'S3 DB scripts folder path'
  Type: String
  Default: 's3://your-s3-bucket/databases'
```

3. Specificate il percorso Amazon S3 della cartella in cui avete inserito la cartella del file system nel passaggio precedente:

```
S3ApplicationFilePath:
  Description: 'S3 application files folder path'
  Type: String
  Default: 's3://your-s3-bucket/file-system'
```

4. Specificate il percorso Amazon S3 della cartella in cui avete inserito la cartella userdata nel passaggio precedente:

```
S3UserDataPath:
```

```
Description: 'S3 userdata folder path'  
Type: String  
Default: 's3://your-s3-bucket/userdata'
```

5. Specificate anche un percorso Amazon S3 in cui salvare i file dei risultati da utilizzare nel passaggio successivo.

```
S3SaveProducedFilesPath:  
Description: 'S3 path folder to save produced files'  
Type: String  
Default: 's3://your-s3-bucket/post-produced-files'
```

6. Cambia l'ID AMI con quello corretto ottenuto in precedenza in questo tutorial utilizzando il seguente modello:

```
BaaAmiId:  
Description: 'ami id (AL2) for ba anywhere'  
Default: 'ami-0bd41245734fd20d9'  
Type: String
```

- Facoltativamente, è possibile modificare il nome delle tre istantanee che verranno create dall'esecuzione dei database di caricamento. withCloudFormation Queste saranno visibili nello CloudFormation stack durante la creazione e verranno utilizzate più avanti in questo tutorial. Ricorda di annotare i nomi usati per le istantanee del database.

```
SnapshotPrimary:  
Description: 'Snapshot Name DB BA Primary'  
Type: String  
Default: 'snapshot-primary'  
  
SnapshotBluesam:  
Description: 'Snapshot Name DB BA Bluesam'  
Type: String  
Default: 'snapshot-bluesam'  
  
SnapshotJics:  
Description: 'Snapshot Name DB BA Jics'  
Type: String  
Default: 'snapshot-jics'
```

**Note**

In questo documento, si presume che il nome delle istantanee rimanga coerente.

7. Eseguilo CloudFormation con CLI o AWS console utilizzando il pulsante Create Stack e la procedura guidata. Al termine del processo, dovresti vedere tre istantanee nella console RDS con il nome che hai scelto seguito da un ID univoco. Questi nomi ti serviranno nel passaggio successivo.

**Note**

RDS aggiungerà dei suffissi ai nomi delle istantanee definiti nel modello. AWS CloudFormation Assicurati di ottenere il nome completo dello snapshot da RDS prima di procedere al passaggio successivo.

Comando CLI di esempio-

```
aws cloudformation create-stack --stack-name load-and-create-ba-snapshots --  
template-url https://your-apptest-bucket.s3.us-west-2.amazonaws.com/load-and-  
create-ba-snapshots.yml --capabilities CAPABILITY_NAMED_IAM
```

Puoi anche verificare nel percorso Amazon S3 che hai fornito per S3 SaveProducedFilePath che i set di dati siano stati creati correttamente.

## Fase 5: Avvia AWS il runtime Blu Age CloudFormation

Utilizzalo CloudFormation per eseguire l'istanza Amazon EC2 con l'applicazione CardDemo AWS Blu Age. È necessario sostituire alcune variabili nel file CloudFormation named `m2-with-ba-using-snapshots-https-authentication.yml` modificando il file YAML o modificando i valori nella console durante l'avvio del CFN.

1. Modifica `AllowedVpcEndpointPrincipals` per specificare quale account raggiungerà l'endpoint VPC per accedere al runtime AWS Blu Age, utilizzando i seguenti comandi:

```
AllowedVpcEndpointPrincipals:  
  Description: 'comma-separated list of IAM users, IAM roles, or AWS accounts'
```

```
Default: 'apptest.amazonaws.com'  
Type: String
```

2. Modifica il valore delle variabili SnapshotPrimaryDb e SnapshotBlusamDb SnapshotJicsDb il nome delle istantanee. Ottieni anche i nomi delle istantanee da RDS dopo averle create nel passaggio precedente.

```
SnapshotPrimary:  
  Description: 'Snapshot DB cluster for DB Primary'  
  Type: String  
  Default: 'snapshot-primary87d067b0'  
  
SnapshotBluesam:  
  Description: 'Snapshot DB cluster for DB Bluesam'  
  Type: String  
  Default: 'snapshot-bluesam87d067b0'  
  
SnapshotJics:  
  Description: 'Snapshot DB cluster for DB Jics'  
  Type: String  
  Default: 'snapshot-jics87d067b0'
```

#### Note

RDS aggiungerà il proprio suffisso ai nomi delle istantanee.

3. Fornisci la tua coppia di chiavi Amazon EC2 per l'istanza EC2, utilizzando questo comando:

```
Ec2KeyPair:  
  Description: 'ec2 key pair'  
  Default: 'm2-tests-us-west-2'  
  Type: String
```

4. Fornisci l'ID AMI che hai ottenuto durante il processo di registrazione AMI per la variabile BaaAmild, utilizzando:

```
BaaAmiId:  
  Description: 'ami id (AL2) for ba anywhere'  
  Default: 'ami-0d0fafcc636fd1e6d'  
  Type: String
```

5. Fornisci il percorso della cartella Amazon S3 che hai utilizzato nel passaggio precedente per salvare i file prodotti, utilizzando il seguente comando:

```
S3ApplicationFilePath:  
  Description: 'bucket name'  
  Type: String  
  Default: 's3://your-s3-bucket/post-produced-files'
```

6. Infine, fornisci il percorso della cartella di s3 -: userdata-folder-path

```
S3UserDataPath:  
  Description: 'S3 userdata folder path'  
  Type: String  
  Default: 's3://your-s3-bucket/userdata'
```

- (Facoltativo) È possibile abilitare la modalità HTTPS e l'autenticazione HTTP di base per tomcat. Sebbene funzionino anche le impostazioni predefinite.

#### Note

Per impostazione predefinita, la modalità HTTPS è disabilitata e impostata sulla modalità HTTP nel parametro BacHttpsMode:

Per esempio:

```
BacHttpsMode:  
  Description: 'http or https for Blue Age Runtime connection mode '  
  Default: 'http'  
  Type: String  
  AllowedValues: [http, https]
```

- (Facoltativo) Per abilitare la modalità HTTPS, è necessario modificare il valore in HTTPS e fornire l'ARN del certificato ACM modificando il valore della variabile ACM: CertArn

```
ACMCertArn:  
  Type: String  
  Description: 'ACM certificate ARN'  
  Default: 'your arn certificate'
```

- (Facoltativo) L'autenticazione di base è disabilitata per impostazione predefinita con il parametro `WithBacBasicAuthentication` impostato su `false`. È possibile abilitarla impostando il valore su `true`.

```
WithBacBasicAuthentication:  
  Description: 'false or true for Blue Age Runtime Basic Authentication '  
  Default: false  
  Type: String  
  AllowedValues: [true, false]
```

7. Una volta completata la configurazione, è possibile creare lo stack utilizzando il CloudFormation modello modificato.

## Fase 6: Test dell'AWSistanza Amazon EC2 Blue Age

Esegui manualmente il CloudFormation modello per creare l'istanza AWS Blu Age Amazon EC2 per l' `CardDemo` applicazione per assicurarti che si avvii senza errori. Questa operazione viene eseguita per verificare che il CloudFormation modello e tutti i prerequisiti siano validi, prima di utilizzare il CloudFormation modello con la funzionalità Application Testing. Puoi quindi utilizzare Application Testing per creare automaticamente l'istanza di destinazione di Amazon EC2 AWS Blu Age durante la riproduzione e confrontarla con una condizione iniziale.

1. Esegui il comando CloudFormation `create stack` per creare l'istanza AWS Blu Age Amazon EC2, fornendo il modello `m2 with-ba-using-snapshots - CloudFormation -https-authentication.yml` che hai modificato nel passaggio precedente:

```
aws cloudformation create-stack --stack-name load-and-create-ba-snapshots --  
template-url https://apptest-ba-demo.s3.us-west-2.amazonaws.com/m2-with-ba-using-  
snapshots-https-authentication.yml --capabilities CAPABILITY_NAMED_IAM --region us-  
west-2
```

### Note

Ricorda di specificare la regione corretta in cui è stata ripristinata l'AMI AWS Blu Age.

2. Assicurati che tutto funzioni correttamente cercando nella console l'istanza Amazon EC2 in esecuzione. Connect ad esso utilizzando Session Manager.
3. Dopo esserti connesso all'istanza Amazon EC2, utilizza i seguenti comandi:

```
sudo su
cd /m2-anywhere/tomcat.gapwalk/velocity/logs
cat catalina.log
```

4. Assicurati che non vi siano eccezioni o errori nel registro.
5. Quindi, verifica che l'applicazione risponda utilizzando questo comando:

```
curl http://localhost:8080/gapwalk-application/
```

Verrà visualizzato il messaggio «L'applicazione Jics è in esecuzione».

## Passaggio 7: verifica che i passaggi precedenti siano stati completati correttamente

Nei prossimi passaggi, utilizzeremo AWS Mainframe Modernization Application Testing per riprodurre e confrontare i set di dati creati dall'applicazione. CardDemo Questi passaggi si basano sul completamento con successo di tutti i passaggi precedenti di questo tutorial. Convalida quanto segue prima di procedere:

1. Hai creato con successo l'istanza AWS Blu Age on Amazon EC2 tramite il AWS CloudFormation modello.
2. Il servizio Tomcat su AWS Blu Age su Amazon EC2 è attivo e funzionante, senza eccezioni.

Quando l'istanza EC2 viene eseguita con l' CardDemo applicazione, completa i seguenti passaggi sulla console Application Testing per eseguire la riproduzione e il confronto dei set di dati in batch.

## Fase 8. Crea una condizione iniziale

In questo passaggio, crei una condizione iniziale fornendo il CloudFormation modello utilizzato per distribuire l' CardDemo applicazione AWS Blu Age su Amazon EC2.

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/.](https://console.aws.amazon.com/m2/)
2. Nel riquadro di navigazione a sinistra, scegli Application Testing.
3. In Application Testing, scegli Crea condizione iniziale.

4. Usa la tua copia locale con percorsi Amazon S3 modificati e ID snapshot che puntano alle tue risorse.

## Fase 9: Creare il test case

In questo passaggio, crei il test case che verrà utilizzato per confrontare i set di dati creati nell'applicazione Card Demo.

1. Crea un nuovo test case. Dategli un nome e una descrizione.
2. Specificare CREAMTMT .JCL come nome JCL.
3. Aggiungi i seguenti set di dati alla definizione del test case:

Nome	CCSID	RecordFormat	RecordLength
AWS.M2.CA RDDEMO.ST ATEMNT.PS	«037"	FB	80
AWS.M2.CA RDDEMO.ST ATEMENT.HTML	«037"	FB	100

### Note

Il nome JCL e i dettagli del set di dati devono corrispondere.

## Fase 10: Creare uno scenario di test

1. Crea un nuovo scenario di test e forniscine un nome e una descrizione.
2. Aggiungi il test case creato nel passaggio precedente allo scenario di test.
3. Una volta creato lo scenario di test, seleziona la condizione iniziale creata nel passaggio 1 nella pagina di panoramica dello scenario di test.



## Fase 11: Registra lo scenario di test

In questo passaggio, esegui i casi di test sul codice sorgente. Per farlo:

1. Scarica ed esegui i set di dati che hanno avuto origine dall'esecuzione mainframe dell'applicazione. CardDemo
2. Carica la cartella decompressa nel tuo bucket Amazon S3. Questo bucket Amazon S3 deve trovarsi nella stessa regione delle altre risorse per il test delle applicazioni.

### Note

Dovrebbero esserci due file con i nomi corrispondenti ai nomi dei set di dati esaminati nel test case precedente.

3. Nella pagina di panoramica dello scenario di test, scegli il pulsante Registra.
4. Nella pagina di registrazione dello scenario di test, specifica la posizione Amazon S3 in cui hai caricato i set di dati ottenuti dal mainframe di origine.
5. Fai clic su Invia per avviare il processo di registrazione.

### Note

Attendi il completamento della registrazione prima di eseguire la riproduzione e il confronto.

## Passaggio 12: Riproduci e confronta

Esegui lo scenario di test e i casi di test nell'ambiente AWS AWS Blue Age on Amazon EC2 di destinazione. Application Testing acquisirà i set di dati prodotti dai replay e li confronterà con i set di dati di riferimento registrati sul mainframe.

1. Scegli Replay and Compare. Dovrebbero essere necessari circa tre minuti per creare lo CloudFormation stack, eseguire il confronto ed eliminare lo stack.

Una volta completato il tutto, dovresti avere risultati di confronto con alcune differenze create intenzionalmente allo scopo di questa demo.

## AWS Pagine di codici per i set di dati supportati dai test delle applicazioni di modernizzazione del mainframe

AWS Application Testing è in versione di anteprima per AWS Mainframe Modernization ed è soggetto a modifiche. Si consiglia di utilizzare questa funzionalità solo con dati e applicazioni di test e non in ambienti di produzione.

Utilizzate la tabella seguente per determinare se l'identificatore del set di caratteri codificato (CCSID) per i vostri dati è supportato in Application Testing. AWS [Se i tuoi dati utilizzano un CCSID non supportato, ti consigliamo di convertirlo in un CCSID supportato o di contattarci per ricevere assistenza.](#)

CCSID	Set di caratteri	Descrizione
37	IBM037, IBM-037, Cp037	Ospite: USA, Canada (ESA), Paesi Bassi, Portogallo, Brasile, Australia, Nuova Zelanda
273	IBM 273, IBM-273, Cp273	Organizzatore: Austria, Germania
277	IBM 277, IBM-277, Cp277	Organizzatore: Danimarca, Norvegia
278	IBM 278, IBM-278, Cp278	Organizzatore: Finlandia, Svezia
280	IBM 280, IBM-280, Cp280	Ospite: Italia
284	IBM284, IBM-284, Cp284	Organizzatore: Spagna, America Latina (spagnolo)
285	IBM285, IBM-285, Cp285	Organizzatore: Regno Unito
297	IBM297, IBM-297, Cp297	Organizzatore: Francia

CCSID	Set di caratteri	Descrizione
300	IBM-300	JAPAN DB EBCDIC
301	IBM-301	Dati del PC: Japan DB
437	IBM437, IBM-437, US-ASCII, ASCII, Cp437, US-ASCII	Dati PC: PC Base USA, molti altri paesi
500	IBM 500, IBM-500, Cp500	Organizzatore: Belgio, Canada (AS/400), Svizzera, International Latin-1
720	IBM-720	MSDOS ARABO
737	IBM-737, x-IBM 737	MSDOS GRECO
775	IBM 775, IBM-775	MSDOS BALTICO
808	IBM-808	Dati del PC: cirillico, Russia, con euro
813	ISO-8859-7, ISO8859_7	ISO 8859-7: Grecia
819	ISO-8859-1, ISO8859_1	ISO 8859-1: paesi latino-1
833	IBM-833	EBCDIC COREANO
834	IBM-834, x-IBM834	DB EBCDIC COREANO
835	IBM-835	T-CHINESE DB EBCD
836	IBM-836	EBCDIC S-CINESE
837	IBM-837	EBCDIC S-CINESE
850	IBM 850, IBM-850, Cp 850	Dati PC: Paesi Latin-1
855	IBM855, IBM-855, Cp855	Dati del PC: cirillico
856	IBM-856, X-IBM 856, Cp856	Dati del PC: ebraico

CCSID	Set di caratteri	Descrizione
858	IBM00858, IBM-858, Cp858	Dati PC: paesi latino-1, con euro
859	IBM-859	Dati del PC: LATIN-9
860	IBM860, IBM-860	Dati del PC: portoghese
861	IBM861, IBM-861	Dati PC: Islanda
862	IBM 862, IBM-862, Cp 862	Dati PC: ebraico (migrazione)
863	IBM863, IBM-863	Dati PC: Canada
865	IBM865, IBM-865, Cp865	Dati del PC: Den/Norway
866	IBM866, IBM-866, Cp866	Dati del PC: cirillico, Russia
867	IBM-867	Dati del PC: ebraico con euro
870	IBM 870, IBM-870, Cp 870	Host: Latin-2 multilingue
871	IBM871, IBM-871, Cp871	Host: Islanda
874	x-IBM 874	Dati del PC: thailandese
875	IBM-875, x-IBM875, Cp875	Host: Grecia
897	IBM-897	Dati del PC: Japan SB
912	ISO-8859-2, ISO8859_2	ISO 8859-2: Latin-2 multilingue
915	ISO-8859-5, ISO8859_5	ISO 8859-5: cirillico
916	ISO-8859-8, ISO8859_8	ISO 8859-8: ebraico
918	IBM 918, IBM-918, Cp 918	Ospite: Urdu
920	ISO-8859-9, ISO8859_9	ISO 8859-9: Latin-5 (ECMA-128, Turchia TS-5881)

CCSID	Set di caratteri	Descrizione
921	IBM-921, X-IBM 921, Cp 921	Dati PC: Lettonia, Lituania
922	IBM-922, x-IBM 922, Cp 922	Dati PC: Estonia
923	ISO-8859-15, Cp923, ISO8859_15_FDIS	ISO 8859-15: Latino-9
924	IBM-924	ISO 8859-15: Latino-9
927	IBM-927	Dati del PC: T-Chinese
930	IBM-930, X-IBM 930, Cp 930	Katakana Host: SBCS esteso. Kanji Host: DBCS che include 4370 caratteri definiti dall'utente
932	IBM-932	Dati del PC: Japan Mix
933	IBM-933, X-IBM 933, Cp 933	Host: SBCS esteso. Host: DBCS che include 1880 caratteri definiti dall'utente e 11172 caratteri Hangul completi
935	IBM-935, X-IBM 935, Cp 935	Host: SBCS esteso. Host: DBCS che include 1880 caratteri definiti dall'utente.
937	IBM-937, X-IBM 937, Cp 937	Host: SBCS esteso. Host: DBCS che include 6204 caratteri definiti dall'utente
939	IBM-939, X-IBM 939, Cp 939	Latin Host: SBCS esteso. Kanji Host: DBCS che include 4370 caratteri definiti dall'utente.

CCSID	Set di caratteri	Descrizione
942	IBM-942, IBM-942 C, X-IBM 942, X-IBM942 C, Cp 942, Cp942 C	Dati PC: SBCS esteso. Dati PC: DBCS che include 1880 caratteri definiti dall'utente
943	IBM-943, IBM-943C, Shift_JIS , Windows-31j, Windows-932, X-IBM943, X-IBM943c, Cp943, Cp943C, MS932	Dati del PC: SBCS. Dati PC: DBCS per ambiente aperto, inclusi 1880 caratteri IBM definiti dall'utente
947	IBM-947	T-CHINESE BIG-5
948	IBM-948, X-IBM 948, Cp 948	Dati PC: SBCS esteso. Dati PC: DBCS che include 6204 caratteri definiti dall'utente
949	IBM-949, IBM-949 C, X-IBM 949, X-IBM 949 C, Cp 949 C	Codice IBM KS - Dati del PC: SBCS. Codice IBM KS - Dati PC: DBCS, inclusi 1880 caratteri definiti dall'utente
950	Big5, IBM-950, x-IBM 950, Cp950	Dati del PC: SBCS (IBM BIG5). Dati PC: DBCS incluso 13493 CNS, 566 IBM selezionati, 6204 caratteri definiti dall'utente
951	IBM-951	Dati del PC: IBM KS
954	EUC-JP, IBM-954, IBM-954C	G0: JIS X201 romano. G1: È X208-1990. G1: JIS X201 Katakana. G1: JIS X212
964	EUC-TW, IBM-964, X-IBM964, Cp964	G0: ASCII. G1: CNS 11643 piano 1. G1: CNS 11643 piano 2.

CCSID	Set di caratteri	Descrizione
970	EUC-KR, x-IBM 970, Cp970	G0: ASCII. G1: KSC X5601-1989 inclusi 1880 caratteri definiti dall'utente
971	IBM-971	EUC COREANO
1006	IBM-1006, X-IBM 1006, Cp1006	ISO-8: urdu
1025	IBM-1025, X-IBM 1025, Cp 1025	Host: cirillico multilingue
1026	IBM 1026, IBM-1026, Cp 1026	Host: Latin-5 (Turchia)
1027	IBM-1027	EBCD LATINO GIAPPONESE
1041	IBM-1041	Dati del PC: Giappone
1043	IBM-1043	Dati del PC: T-Chinese
1046	IBM-1046, IBM-1046S, x-IBM1046	ARABO - PC
1047	IBM1047, IBM-1047	Ospite: Latin-1
1051	hp - Roman 8	EMULAZIONE HP
1088	IBM-1088	Dati del PC: Korea KS
1089	ISO-8859-6, ISO8859_6	ISO 8859-6: arabo
1097	IBM-1097, x-IBM1097, Cp1097	Ospite: Farsi
1098	IBM-1098, x-IBM 1098, Cp 1098	Dati del PC: persiano
1112	IBM-1112, x-IBM 1112, Cp 1112	Ospite: Lettonia, Lituania

CCSID	Set di caratteri	Descrizione
1114	IBM-1114	Dati del PC: T-CH SB
1115	IBM-1115	Dati del PC: S-CH GB
1122	IBM-1122, x-IBM1122, Cp1122	Ospite: Estonia
123	IBM-1123, x-IBM 1123, Cp 1123	Ospite: Ucraina cirillica
1124	IBM-1124, x-IBM 1124, Cp 1124	8-bit: cirillico, Bielorussia
1140	IBM 01140, IBM-1140, Cp 1140	Organizzatore: USA, Canada (ESA), Paesi Bassi, Portogallo, Brasile, Australia, Nuova Zelanda, con euro
1141	IBM01141, IBM-1141, Cp 1141	Organizzatore: Austria, Germania, con euro
1142	IBM01142, IBM-1142, Cp 1142	Organizzatore: Danimarca, Norvegia, con euro
1143	IBM 01143, IBM-1143, Cp 1143	Organizzatore: Finlandia, Svezia, con euro
1144	IBM0114, IBM-1144, Cp1144	Organizzatore: Italia, con euro
1145	IBM 01145, IBM-1145, Cp 1145	Organizzatore: Spagna, America Latina (spagnolo), con euro
1146	IBM01146, IBM-1146, Cp 1146	Organizzatore: Regno Unito, con euro
1147	IBM 01147, IBM-1147, Cp 1147	Organizzatore: Francia, con euro



CCSID	Set di caratteri	Descrizione
1148	IBM 01148, IBM-1148, Cp 1148	Organizzatore: Belgio, Canada (AS/400), Svizzera, International Latin-1, con euro
1149	IBM 01149, IBM-1149, Cp 1149	Organizzatore: Islanda, con euro
1200	UTF-16BE	Unicode con set di caratteri 65535. In assenza di un byte-order mark (BOM), si presume che sia UTF-16 BE (big-endian).
1202	UTF-16LE	UTF-16 LE con IBM PUA
1204	UTF-16	UTF-16 con IBM PUA
1208	UTF-8, UTF-8J, UTF8	Unicode con set di caratteri 65535. UTF-8.
1232	UTF-32BE	UTF-32 BE con IBM PUA
1234	UTF-32LE	UTF-32 LE con IBM PUA
1236	UTF-32	UTF-32 con IBM PUA
1351	IBM-1351	GIAPPONE APERTO
1362	IBM-1362	MS-WIN COREANO
1363	IBM-1363, IBM-1363C, Windows-949, MS949	Dati del PC: MS Windows Korean SBCS. Dati del PC: MS Windows Korean DBCS incluso 11172 Hanguk completo

CCSID	Set di caratteri	Descrizione
1364	IBM-1364	Ospite: SBCS esteso. Host: DBCS che include 1880 caratteri definiti dall'utente e 11172 caratteri Hangul completi
1370	IBM-1370	Dati PC: SBCS esteso, con euro. Dati PC: DBCS che include 6204 caratteri definiti dall'utente, con euro
1371	IBM-1371	Organizzatore: Extended SBCS, con euro. Host: DBCS che include 6204 caratteri definiti dall'utente, con euro
1375	Big 5-HKSC	Big-5 Ext misto per HKSCS
1380	IBM-1380	Dati del PC: S-CH GB
1381	IBM-1381, x-IBM 1381, Cp 1381	Dati del PC: SBCS esteso (IBM GB). Dati PC: DBCS (IBM GB), inclusi 31 caratteri selezionati da IBM e 1880 caratteri definiti dall'utente
1382	IBM-1382	EUC S-CINESE
1383	EUC-CN, GB2312, IBM-1383, x-IBM1383, Cp1383	G0: ASCII. G1: set GB 2312-80
1385	IBM-1385	Dati del PC: S-CH GBK
1386	GBK, IBM-1386, Windows-936, MS936	Dati PC: GBK S-cinese e IBM BIG-5 T-cinese. Dati del PC: GBK S-Chinese

CCSID	Set di caratteri	Descrizione
1388	IBM-1388	Ospite: SBCS esteso. Host: DBCS che include 1880 caratteri definiti dall'utente
1390	IBM-1390	Katakana Host: SBCS esteso, con euro. Kanji Host: DBCS che include 6205 caratteri definiti dall'utente
1399	IBM-1399	Latin Host: SBCS esteso, con euro. Kanji Host: DBCS che include 4370 caratteri definiti dall'utente, con euro
5050	JIS0201, JIS0208, JIS0212, JIS0201, JIS0208, JIS0212	G0: JIS X201 ROMAN. G1: È X208-1990. G1: JIS X201 Katakana. G1: JIS X212
5054	ISO-2022-JP	TCP GIAPPONESE
5346	windows-1250, Cp1250	MS Windows: Latin-2, versione 2 con euro
5347	windows-1251, Cp1251	MS Windows: cirillico, versione 2 con euro
5348	windows-1252, Cp1252	MS Windows: paesi Latin-1, versione 2 con euro
5349	windows-1253, Cp1253	MS Windows: Grecia, versione 2 con euro
5350	windows-1254, Cp1254	MS Windows: Turchia, versione 2 con euro
5351	windows-1255, Cp1255	MS Windows: ebraico, versione 2 con euro

CCSID	Set di caratteri	Descrizione
5352	windows-1256, Windows-1256s, Cp1256	MS Windows: arabo, versione 2 con euro
5353	windows-1257, Cp1257	MS Windows: Baltic Rim, versione 2 con euro
5354	windows-1258, Cp1258	MS Windows: vietnamita, versione 2 con euro
5488	GB18030	GB18030, dati a 1 byte GB18030, dati a 2 byte GB18030, dati a 4 byte
9030	IBM-838, Cp838	Ospite: Thai Extended SBCS
9066	IBM-874, Cp874	Dati PC: SBCS esteso thailandese
9400	CESU-8	CESU-8 con IBM PUA
25546	ISO-2022-IT	TCP COREANO
33722	IBM-33722, IBM-33722C	IBM eUCJP

# Trasferimento di file nella modernizzazione del AWS mainframe

AWS Mainframe Modernization File Transfer consente di trasferire e convertire set di dati mainframe in Amazon S3 per casi d'uso di modernizzazione, migrazione e potenziamento del mainframe.

## Argomenti

- [Cos'è AWS Mainframe Modernization File Transfer?](#)
- [Installare un agente di trasferimento file](#)
- [Endpoint di trasferimento dati](#)
- [Trasferisci attività](#)
- [Tutorial: Come iniziare a usare AWS Mainframe Modernization File Transfer](#)

## Cos'è AWS Mainframe Modernization File Transfer?

Con AWS Mainframe Modernization File Transfer, puoi trasferire e convertire set di dati e file con un servizio completamente gestito per accelerare e semplificare i casi d'uso di modernizzazione, migrazione e potenziamento verso il servizio di AWS modernizzazione del mainframe e Amazon S3.

## Argomenti

- [Vantaggi del trasferimento di file per la modernizzazione dei mainframe di AWS](#)
- [Come funziona AWS Mainframe Modernization File Transfer](#)

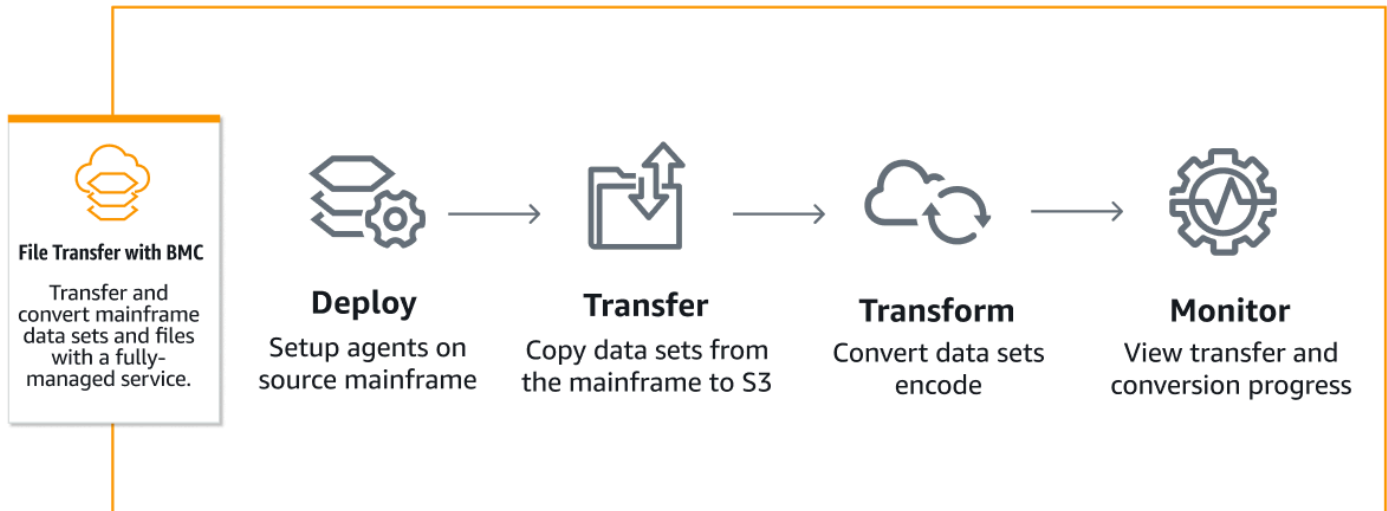
## Vantaggi del trasferimento di file per la modernizzazione dei mainframe di AWS

AWS Mainframe Modernization File Transfer ti aiuta a trasferire set di dati dal mainframe ad Amazon S3. Alcuni vantaggi includono:

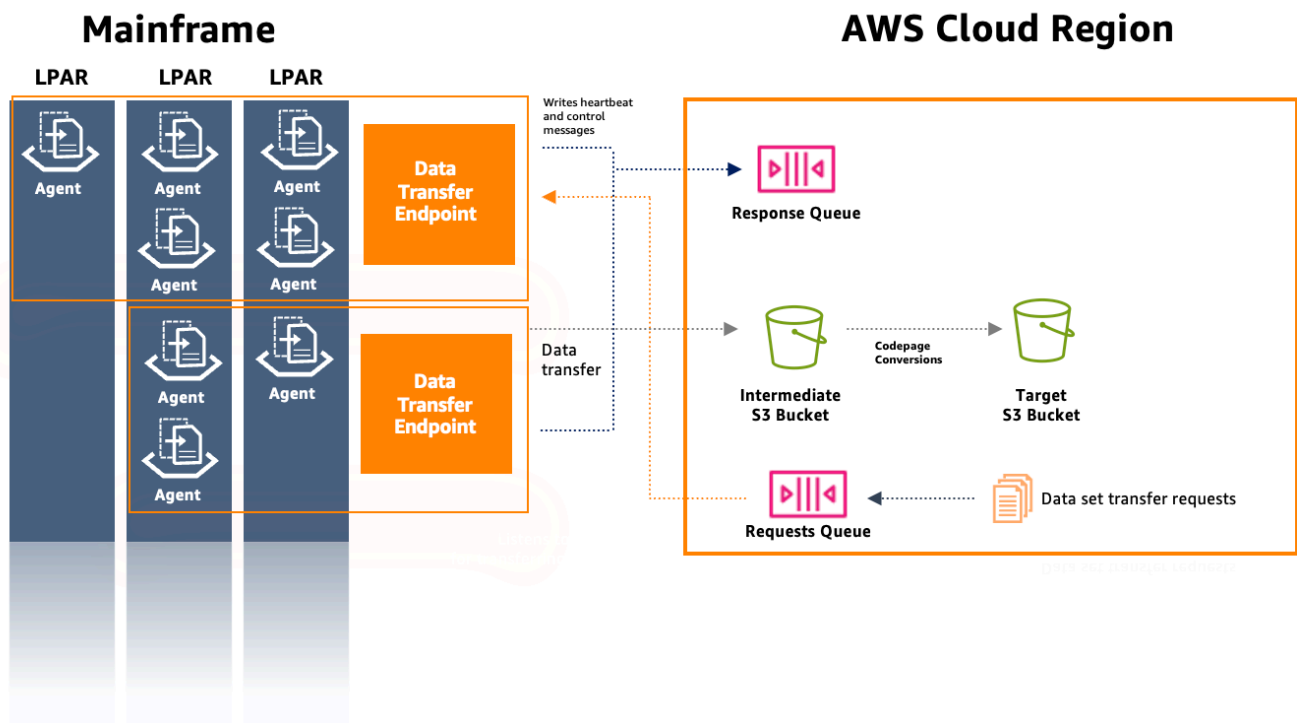
- Scoperta dei set di dati e degli artefatti mainframe di origine
- Trasferimenti automatici e conversione di set di dati
- Scalabilità, efficienza e velocità per ottenere trasferimenti di set di dati più rapidi verso AWS

# Come funziona AWS Mainframe Modernization File Transfer

La figura seguente è una panoramica di come funziona AWS Mainframe Modernization File Transfer a livello concettuale.



La figura seguente è una panoramica dell'architettura della funzionalità AWS Mainframe Modernization File Transfer.



# Installare un agente di trasferimento file

Segui questa guida step-by-step per completare i prerequisiti per l'installazione di un agente sul mainframe di origine e per configurare l'agente.

## Argomenti

- [Fase 1: Accedere a ISPF](#)
- [Fase 2: Allocare un set di dati per z/FS](#)
- [Passaggio 3: Formattare il set di dati come z/FS](#)
- [Fase 4: Definire il filesystem su z/OS](#)
- [Fase 5: Montare il file system](#)
- [Fase 6: Verificare il montaggio](#)
- [Fase 7: Inserire gli OMV](#)
- [Passaggio 8: Impostare la variabile di ambiente della directory di installazione dell'agente](#)
- [Fase 9: Impostare la variabile di ambiente della directory di lavoro](#)
- [Fase 10: Creare la cartella di lavoro](#)
- [Fase 11: Copiare il pacchetto tar di modernizzazione del AWS mainframe nella directory di lavoro su z/OS](#)
- [Passaggio 12: presupponiamo l'utente root](#)
- [Configura autorizzazioni e STC](#)
- [Crea un utente IAM con credenziali di accesso a lungo termine](#)
- [Crea un ruolo IAM che l'agente deve assumere](#)
- [Configurazione dell'agente](#)

## Fase 1: Accedere a ISPF

Accedi alla tua sessione ISPF (Interactive System Productivity Facility). Questo di solito viene fatto tramite un emulatore di terminale 3270.

## Fase 2: Allocare un set di dati per z/FS

Utilizzando l'utilità per il set di dati di ISPF, allocate un nuovo set di dati per z/FS. In genere, questa operazione viene eseguita nell'utilità di elenco dei set di dati nel passaggio 1.

1. Vai all'opzione 3.4 (Utilità --> Dataset).
2. Premi il tasto 'C' per creare un nuovo set di dati.
3. Immettete il nome del set di dati (ad esempio, 'yourHLQ.M2Agent.zfs').
4. Specificate il tipo di set di dati come «Grande formato» con una dimensione principale di 1000 cilindri e una dimensione secondaria di 200.
5. Imposta l'organizzazione del set di dati (DSORG) come PS e il formato di registrazione (RECFM) come «U» (non definito).
6. Completa il processo di creazione.

## Passaggio 3: Formattare il set di dati come z/FS

Dopo aver creato il set di dati, formattalo come file system z/FS.

Un modo per farlo è utilizzare il seguente Job Control Language (JCL):

```
//FORMAT EXEC PGM=IOEAGFMT,PARM='AGGRNAME(yourhlq.M2AGENT.ZFS),FORMAT,AGGRSIZE(1200)'  
//SYSPRINT DD SYSOUT=A
```

Invia questo lavoro e controlla se è stato completato con successo.

## Fase 4: Definire il filesystem su z/OS

Definire da z/FS a z/OS utilizzando il comando o tramite un processo batch. DEF FILESYSTEM

Utilizza il seguente comando:

```
DEF FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(R/W) MOUNTPPOINT('/usr/lpp/aws/m2-agent')
```

### Note

Questo passaggio richiede l'autorità a livello di sistema.

## Fase 5: Montare il file system

Per montare il filesystem, usa il comando. MOUNT È possibile montare il filesystem nella riga di comando in ISPF o in batch.



Per esempio:

```
MOUNT FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/usr/lpp/aws/m2-agent')
```

## Fase 6: Verificare il montaggio

Verificate che il filesystem sia montato correttamente usando il D OMVS, A comando o controllando all'interno di Unix System Service (USS).

## Fase 7: Inserire gli OMV

Usa il seguente comando per inserire OMV:

```
TSO OMVS
```

## Passaggio 8: Impostare la variabile di ambiente della directory di installazione dell'agente

Utilizzate il comando seguente per impostare l'ambiente della directory di installazione dell'agente:

```
export AGENT_DIR=/usr/lpp/aws/m2-agent
```

## Fase 9: Impostare la variabile di ambiente della directory di lavoro

Utilizzate il seguente comando per impostare la variabile di ambiente della directory di lavoro:

```
export WORK_DIR=$AGENT_DIR/tmp
```

## Fase 10: Creare la cartella di lavoro

Utilizzate il seguente comando per impostare l'ambiente della directory di lavoro:

```
mkdir -p $WORK_DIR
```

## Fase 11: Copiare il pacchetto tar di modernizzazione del AWS mainframe nella directory di lavoro su z/OS

Quando utilizzate FTP o qualsiasi altro metodo di trasferimento, assicuratevi che il file tar sia trasferito in modalità binaria.

### Passaggio 12: presupponiamo l'utente root

Usa il seguente comando per assumere l'utente root:

```
su
```

Segui questi passaggi per completare l'installazione dell'agente:

#### Note

È necessario assumere l'utente root prima di continuare con questi passaggi.

1. Imposta la variabile di ambiente della versione m2-agent sulla versione attualmente in fase di installazione utilizzando il seguente comando:

```
export M2_AGENT_VERSION=1.0.0
```

2. Estrai il pacchetto agent tar usando il seguente comando:

```
tar -xpf m2-agent-package-$M2_AGENT_VERSION.tar -C $AGENT_DIR
```

3. Create un collegamento `current-version` simbolico alla directory di installazione dell'agente corrente con il seguente comando:

```
ln -s $AGENT_DIR/m2-agent-v$M2_AGENT_VERSION $AGENT_DIR/current-version
```

4. Aggiorna e invia CPY#PDS per creare i set di dati dell'agente File Transfer.

#### Note

JCL utilizza l'HLQ SYS2.AWS.M2.

Per creare l'agente File Transfer, imposta le righe dei parametri 000006-000012. Inoltre, aggiorna le tre variabili HLQ simboliche e utilizzale VOLSER successivamente AGNTPATH in JCL:

```
oedit $AGENT_DIR/current-version/installation/CPY#PDS
submit $AGENT_DIR/current-version/installation/CPY#PDS
```

### Note

Questo JCL è personalizzato per configurare alcuni aspetti dell'installazione dell'agente sul mainframe. Alloca i set di dati necessari e quindi copia file specifici dal file system Unix a questi set di dati.

## Configura autorizzazioni e STC

1. Aggiorna e invia uno dei seguenti `SYS2.AWS.M2.SAMPLIB(SEC#RACF)` (per configurare le autorizzazioni RACF) o `SYS2.AWS.M2.SAMPLIB(SEC#TSS)` (per impostare le autorizzazioni TSS) in base alle relative istruzioni. Questi membri sono stati creati nel passaggio precedente.  
`CPY#PDS`
2. Aggiorna l'esportazione PWD in `SYS2.AWS.M2.SAMPLIB(M2AGENT) STC JCL`, se il percorso di directory predefinito dell'agente di trasferimento file (`/usr/lpp/aws/m2-agent`) è stato modificato.
3. Aggiorna e copia il JCL in `SYS2.AWS.M2.SAMPLIB(M2AGENT).SYS1.PROCLIB`
4. Aggiungete `SYS2.AWS.M2.LOADLIB` all'elenco APF usando il seguente comando:

```
SETPROG APF ADD DSNAME(SYS2.AWS.M2.LOADLIB) SMS
```

5. Imposta il gruppo e il proprietario dell'agente `logs` e `diag` delle cartelle sull'utente/gruppo dell'agente (`M2USER/M2GROUP`). Utilizza il seguente comando:

```
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/logs
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/diag
```

## Crea un utente IAM con credenziali di accesso a lungo termine

È necessario creare l'agente mainframe richiesto da un utente IAM per utilizzare le code di risposta e richiesta e salvare i set di dati nei bucket Amazon S3.

Durante la creazione di questo utente:

1. Scegli Allega politiche direttamente nelle opzioni di autorizzazione.
2. Una volta creato l'utente, apri la scheda Credenziali di sicurezza e crea una chiave di accesso. Per ulteriori informazioni sulla creazione della chiave di accesso IAM, consulta [Gestione delle chiavi di accesso per gli utenti IAM](#).
3. Nella sezione Chiave di accesso, seleziona Altro quando richiesto per il caso d'uso.

### Note

Salva la chiave di accesso e la chiave di accesso segreta visualizzate nell'ultima pagina della procedura guidata per la creazione della chiave di accesso, prima di scegliere Fine. Queste chiavi vengono utilizzate per configurare l'agente mainframe.

### Note

Salva l'ARN dell'utente IAM utilizzato per impostare una relazione di trust con un ruolo IAM.

## Crea un ruolo IAM che l'agente deve assumere

Crei un nuovo ruolo IAM con una politica di fiducia personalizzata per il tipo di entità affidabile. La policy utilizzerà il seguente modello:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DataTransferEndpointAgentSqsReceive",
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
```

```
        "sqs:ReceiveMessage"
    ],
    "Resource": "<data-transfer-endpoint-request-queue-arn>"
  },
  {
    "Sid": "DataTransferEndpointS3",
    "Effect": "Allow",
    "Action": "s3:PutObject",
    "Resource": "<data-transfer-endpoint-intermediate-bucket-arn>/*"
  },
  {
    "Sid": "DataTransferEndpointAgentSqsSend",
    "Effect": "Allow",
    "Action": "sqs:SendMessage",
    "Resource": "<data-transfer-endpoint-response-queue-arn>"
  },
  {
    "Sid": "DataTransferEndpointAgentKmsDecrypt",
    "Effect": "Allow",
    "Action": "kms:Decrypt",
    "Resource": "<kms-key-id>"
  }
]
}
```

Dove:

- `request-queue-arn` e `response-queue-arn` sono l'ARN della coda di richiesta Amazon SQS creata durante l'inizializzazione dell'endpoint di trasferimento dati.
- `transfer-bucket-arn` è l'ARN del bucket di trasferimento creato in precedenza.

#### Note

Puoi cercare tutti questi valori utilizzando la console AWS.

#### Note

Salva il nome del ruolo, che utilizzerai in seguito per configurare l'agente mainframe.

## Configurazione dell'agente

Per configurare l'agente File Transfer:

1. Accedi a `$AGENT_DIR/current-version/config`.
2. Modifica il file di configurazione dell'agente `application.properties` per aggiungere una configurazione degli ambienti utilizzando il seguente comando:

```
oedit $AGENT_DIR/current-version/config/application.properties
```

Per esempio:

```
agent.environments[0].account-id=<AWS_ACCOUNT_ID>
agent.environments[0].agent-role-name=<AWS_IAM_ROLE_NAME>
agent.environments[0].access-key-id=<AWS_IAM_ROLE_ACCESS_KEY>
agent.environments[0].secret-access-id=<AWS_IAM_ROLE_SECRET_KEY>
agent.environments[0].bucket-name=<AWS_S3_BUCKET_NAME>
agent.environments[0].environment-name=<AWS_REGION>
agent.environments[0].region=<AWS_REGION>
```

Dove:

- `AWS_ACCOUNT_ID` è l'ID dell'account cliente.
- `AWS_IAM_ROLE_NAME` è il nome del ruolo IAM creato [in the section called “Crea un ruolo IAM che l'agente deve assumere”](#).
- `AWS_IAM_ROLE_ACCESS_KEY` è la chiave di accesso dell'utente IAM creato [in the section called “Crea un utente IAM con credenziali di accesso a lungo termine”](#).
- `AWS_IAM_ROLE_SECRET_KEY` è la chiave segreta di accesso per l'utente IAM creato [in the section called “Crea un utente IAM con credenziali di accesso a lungo termine”](#).
- `AWS_S3_BUCKET_NAME` è il nome del bucket di trasferimento creato con l'endpoint di trasferimento dati.
- `AWS_REGION` è la regione in cui si configura l'agente File Transfer.

 Note

Possono esserci diverse sezioni di questo tipo, purché l'indice tra parentesi — [0] — sia incrementato per ciascuna di esse.

È necessario riavviare l'agente per rendere effettive le modifiche.

## Requisiti

1. Quando un parametro viene aggiunto o rimosso, l'agente deve essere arrestato e avviato. Avvia l'agente di trasferimento file utilizzando il seguente comando nella CLI:

```
/S M2AGENT
```

Per arrestare l'agente M2, utilizzate il seguente comando nella CLI:

```
/P M2AGENT
```

2. È possibile fare in modo che l'agente File Transfer venga trasferito su più regioni e account AWS definendo più ambienti.

```
#Region 1
agent.environments[0].account-id=AWS_ACCOUNT_ID
agent.environments[0].agent-role-name=AWS_IAM_ROLE_NAME
agent.environments[0].access-key-id=AWS_IAM_ROLE_ACCESS_KEY
agent.environments[0].secret-access-id=AWS_IAM_ROLE_SECRET_KEY
agent.environments[0].bucket-name=AWS_S3_BUCKET_NAME
agent.environments[0].environment-name=AWS_REGION
agent.environments[0].region=AWS_REGION

#Region 2
agent.environments[1].account-id=AWS_ACCOUNT_ID
agent.environments[1].agent-role-name=AWS_IAM_ROLE_NAME
agent.environments[1].access-key-id=AWS_IAM_ROLE_ACCESS_KEY
agent.environments[1].secret-access-id=AWS_IAM_ROLE_SECRET_KEY
agent.environments[1].bucket-name=AWS_S3_BUCKET_NAME
agent.environments[1].environment-name=AWS_REGION
agent.environments[1].region=AWS_REGION
```

## Endpoint di trasferimento dati

Gli endpoint di trasferimento dati consentono alta disponibilità, scalabilità e gestione semplificata degli agenti sul mainframe di origine. I singoli agenti sono installati su LPAR mainframe e possono essere raggruppati in un endpoint di trasferimento dati. Quando viene effettuata una richiesta di trasferimento di un set di dati, un agente nell'endpoint di trasferimento dati gestirà il trasferimento. Per avviare i trasferimenti di dati, almeno un agente sull'endpoint di trasferimento dati deve essere online.

Questa procedura presuppone che siano stati completati i passaggi indicati in [Configurazione della modernizzazione del AWS mainframe](#) [Configurazione dell'agente di trasferimento file sul](#) mainframe di origine.

### Creare un endpoint per il trasferimento dei dati

Per creare endpoint di trasferimento dati per File Transfer, è necessario seguire questi passaggi nella console di modernizzazione del AWS mainframe.

Per creare un endpoint per il trasferimento dei dati

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nel Regione AWS selettore, scegli la regione in cui desideri trasferire i file dal mainframe a un bucket Amazon S3.
3. Nella pagina Endpoint di trasferimento dati, in Trasferimento file, scegli Crea endpoint di trasferimento dati.
4. Nella pagina Prerequisiti dell'endpoint di trasferimento dati, leggi tutte le istruzioni per assicurarti di aver completato questi passaggi. Una volta confermato, scegli Avanti.
5. Nella pagina Configura l'endpoint di trasferimento dati, aggiungi le informazioni di base per il tuo endpoint di trasferimento dati.
  1. Nella sezione delle informazioni di base, inserisci il nome, la descrizione e la chiave KMS dell'endpoint di trasferimento dati. [Per ulteriori informazioni sulle chiavi KMS, consulta Creare chiavi.](#)



**Note**

Il nome dell'endpoint di trasferimento dati deve corrispondere al nome definito durante la configurazione del file transfer agent sul mainframe di origine.

**Note**

È necessario aggiungere la seguente politica basata sulle risorse per KMS in modo che il servizio AWS Mainframe Modernization possa leggere e utilizzare queste chiavi per la crittografia/decriptografia:

```
{
  "Sid" : "Enable AWS M2 Permissions",
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : [
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Resource" : "*"
}
```

2. Specificate la posizione S3 per i dati intermedi, che è la posizione S3 intermedia in cui vengono archiviati i set di dati trasferiti dal mainframe.

**Note**

Si consiglia di creare un nuovo bucket Amazon S3 per le attività di trasferimento. Per ulteriori informazioni, consulta [Creazione di un bucket](#). Puoi anche sfogliare i bucket Amazon S3 esistenti scegliendo l'opzione Browse S3.

3. Dopo aver inserito i campi obbligatori, scegli Avanti.
6. Nella pagina Rivedi e crea un endpoint per il trasferimento dei dati, controlla se hai completato i prerequisiti e rivedi le informazioni di base. Una volta confermato, scegli Crea e connetti.

7. Nella pagina Verifica connettività, vedrai tutti gli agenti visualizzati con i relativi ID e battiti cardiaci una volta stabilita la connettività dell'agente. Una volta stabilita la connettività, viene visualizzato il messaggio «L'endpoint di trasferimento dati è stato creato correttamente».

#### Note

Se viene visualizzato il messaggio «Impossibile stabilire la connettività dell'agente», rivedi il passaggio 4 per assicurarti di aver completato tutti i prerequisiti richiesti.

8. Scegli Fine.

Verrai reindirizzato alla pagina di panoramica degli endpoint di trasferimento dati, dove potrai vedere l'elenco di tutti gli endpoint di trasferimento dati. Potrai anche vedere gli endpoint di trasferimento dati disponibili o che hanno avuto esito negativo.

Puoi anche cercare gli endpoint di trasferimento dati per nome e accedere a informazioni aggiuntive per ogni agente disponibile.

## Trasferisci attività

Le attività di trasferimento vengono utilizzate per definire la codifica di origine e la codifica di destinazione dei set di dati. La codifica di origine è il formato dei set di dati di origine e la codifica di destinazione è il formato in cui questi set di dati verranno archiviati nel bucket Amazon S3 di destinazione. Questi bucket di destinazione sono definiti dalle attività di trasferimento.

Questa procedura presuppone che tu abbia completato i passaggi [Configurazione della modernizzazione del AWS mainframe](#) e la configurazione. [the section called “Endpoint per il trasferimento dei dati”](#)

### Argomenti

- [Crea attività di trasferimento](#)
- [Visualizza le attività di trasferimento](#)

## Crea attività di trasferimento

Per creare attività di trasferimento per File Transfer, è necessario seguire questi passaggi nella console di modernizzazione del AWS mainframe.

## Per creare un'attività di trasferimento

### Note

È necessario disporre di almeno un endpoint di trasferimento dati per creare nuove attività di trasferimento.

1. [Apri la console di modernizzazione del AWS mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
  2. Nel Regione AWS selettore, scegli la regione in cui desideri trasferire i file dal mainframe a un bucket Amazon S3.
  3. Nella pagina Attività di trasferimento, in Trasferimento file, seleziona un endpoint di trasferimento dati per creare attività di trasferimento.
  4. Se il tuo endpoint di trasferimento dati non dispone di attività di trasferimento, puoi creare nuove attività scegliendo Crea attività di trasferimento.
  5. Nella pagina Crea attività di trasferimento, imposta le proprietà per l'attività di trasferimento.
- In questa pagina, inserisci le informazioni di base dell'operazione di trasferimento, tra cui il nome dell'attività di trasferimento, la descrizione, la chiave segreta e i criteri di ricerca dei set di dati.

### Note

- Crittografa il segreto utilizzando la chiave KMS definita con l'endpoint di trasferimento dati. Il segreto deve contenere anche le credenziali del mainframe necessarie per accedere al set di dati sul mainframe utilizzando le chiavi `and.userId password` Per ulteriori informazioni, consulta [AWS Secrets Manager secret](#).
- È necessario configurare la chiave segreta con la seguente politica basata sulle risorse in modo che il servizio di modernizzazione del AWS mainframe possa accedervi per eseguire attività di trasferimento dei dati:

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

```
    "Service" : "m2.amazonaws.com"
  },
  "Action" : [ "secretsmanager:GetSecretValue",
               "secretsmanager:DescribeSecret" ],
  "Resource" : "*"
} ]
}
```

### Note

La dimensione massima attualmente supportata del set di dati è 90 GiB.

- Inserisci o sfoglia la posizione del bucket Amazon S3 di destinazione per i tuoi file.
  - Verrà selezionato l'endpoint di trasferimento dati predefinito. Puoi anche scegliere di cambiare l'endpoint dagli endpoint disponibili.
6. Seleziona Avanti.
  7. Nella pagina Seleziona set di dati, è necessario aggiornare manualmente la codifica di origine e la codifica di destinazione per ciascuno dei set di dati scelti. La codifica di origine è il formato del set di dati di origine e la codifica di destinazione è il formato del set di dati di destinazione e viene utilizzata per convertire i set di dati.
  8. Dopo aver confermato la codifica di origine e destinazione, scegli Avanti.
  9. Nella pagina Rivedi e crea, puoi rivedere o modificare le informazioni relative all'attività di trasferimento.
  10. Scegli Crea attività di trasferimento.

Viene visualizzato il messaggio «Attività di trasferimento creata con successo».

## Visualizza le attività di trasferimento

Per visualizzare le attività di trasferimento per File Transfer, è necessario seguire questi passaggi nella console di modernizzazione del AWS mainframe.

Per visualizzare le attività di trasferimento

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/.](https://console.aws.amazon.com/m2/)

2. Nel Regione AWS selettore, scegli la regione in cui desideri trasferire i file dal mainframe a un bucket Amazon S3.
3. Nella pagina Attività di trasferimento, in Trasferimento file, seleziona l'endpoint di trasferimento dati per visualizzare le attività di trasferimento.
4. Per gli endpoint con attività di trasferimento preesistenti, queste verranno inserite nella sezione Attività di trasferimento. Puoi scegliere di visualizzare i dettagli di qualsiasi attività di trasferimento da questo elenco.

## Tutorial: Come iniziare a usare AWS Mainframe Modernization File Transfer

AWS Mainframe Modernization File Transfer consente di trasferire e convertire set di dati mainframe per casi d'uso di modernizzazione, migrazione e potenziamento del mainframe.

Segui i passaggi di questo tutorial per capire come funziona AWS Mainframe Modernization File Transfer.

### Panoramica

File Transfer è composto da quanto segue:

1. Un agente da installare sul mainframe di origine.
2. Accesso alle funzionalità di individuazione, trasferimento e conversione dei set di dati direttamente dalla console del servizio di gestione AWS Mainframe Modernization.

Come utente, puoi trasferire set di dati dal mainframe al tuo bucket Amazon S3.

### Argomenti

- [Passaggio 1: trasferire il pacchetto tar dei binari dell'agente dalla partizione logica del mainframe AWS](#)
- [Fase 2: Configurare l'agente File Transfer sul mainframe di origine](#)
- [Fase 3: Creare un endpoint per il trasferimento dei dati](#)
- [Passaggio 4: Creare un'attività di trasferimento](#)
- [Passaggio 5: Visualizza lo stato di avanzamento dell'operazione di trasferimento](#)

## Passaggio 1: trasferire il pacchetto tar dei binari dell'agente dalla partizione logica del mainframe AWS

[Scarica i file tar dal link tar di M2-Agent.](#)

## Fase 2: Configurare l'agente File Transfer sul mainframe di origine

In questa fase, configuri e avvii l'agente AWS Mainframe Modernization File Transfer sul mainframe di origine. L'agente è necessario per facilitare le comunicazioni tra la funzionalità del servizio File Transfer e il mainframe di origine. È richiesto almeno un agente per mainframe. È possibile avviare più di un agente per un'elevata disponibilità e una maggiore scalabilità.

Segui le istruzioni riportate nella [the section called "Installa un agente di trasferimento file"](#) guida per completare l'installazione dell'agente File Transfer sul mainframe.

## Fase 3: Creare un endpoint per il trasferimento dei dati

Segui i passaggi indicati [the section called "Endpoint per il trasferimento dei dati"](#) nella pagina per creare un nuovo endpoint di trasferimento dati.

## Passaggio 4: Creare un'attività di trasferimento

Segui i passaggi indicati [the section called "Trasferisci attività"](#) nella pagina per creare e gestire le attività di trasferimento.

## Passaggio 5: Visualizza lo stato di avanzamento dell'operazione di trasferimento

È possibile visualizzare lo stato di avanzamento dell'attività di trasferimento nella console di modernizzazione del AWS mainframe. Per maggiori dettagli, consulta la sezione [the section called "Visualizza le attività di trasferimento"](#).

# Sicurezza nella modernizzazione del AWS mainframe

Per AWS, la sicurezza del cloud ha la massima priorità. In quanto cliente AWS, è possibile trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle organizzazioni più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra te e AWS. Il [modello di responsabilità condivisa](#) descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che esegue i servizi AWS nel cloud AWS. AWS fornisce inoltre servizi che puoi utilizzare in sicurezza. Revisori di terze parti testano regolarmente e verificano l'efficacia della nostra sicurezza nell'ambito dei [Programmi di conformità AWS](#). Per ulteriori informazioni sui programmi di conformità che si applicano alla modernizzazione del AWS mainframe, consulta [AWS Services in Scope by Compliance Program AWS](#) Program.
- 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 tuoi requisiti aziendali e le leggi e le normative applicabili

Questa documentazione ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi AWS Mainframe Modernization. Mostra come configurare la modernizzazione del AWS mainframe per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse di modernizzazione del AWS mainframe.

AWSMainframe Modernization fornisce le proprie risorse protette dall'IAM (applicazione, ambiente, implementazione, ecc.), che sono le risorse amministrative per la modernizzazione del AWS mainframe, sulle quali qualsiasi azione deve essere consentita dalle politiche IAM.

AWSLa modernizzazione del mainframe per il replatforming è inoltre garantita da IAM. IAM concede o nega l'autorizzazione a un principale per un'azione specifica su una risorsa definita, derivata dall'ambiente mainframe originale, anche tramite policy IAM standard. Il runtime di replatforming di AWS Mainframe Modernization richiama il servizio di autorizzazione IAM quando un'applicazione tenta tale azione su una risorsa protetta. IAM restituirà l'opzione Allow o Deny in base ai meccanismi standard di valutazione delle policy IAM.

Indice

- [Protezione dei dati nella modernizzazione del AWS mainframe](#)
- [Identity and Access Management per la modernizzazione del AWS mainframe](#)
- [Convalida della conformità per la conformità per la conformitàAWS per valida della conformità per la conformità](#)
- [Resilienza inAWSMainframe Modernization](#)
- [Sicurezza dell'infrastruttura in AWS Mainframe Modernization](#)
- [Accesso AWS Mainframe Modernization tramite un endpoint di interfaccia \(\) AWS PrivateLink](#)

## Protezione dei dati nella modernizzazione del AWS mainframe

Il modello di [responsabilità AWS condivisa Il modello](#) di si applica alla protezione dei dati nella modernizzazione mainframe. AWS Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i. Cloud AWS L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Inoltre, sei responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS che utilizzi. Per ulteriori informazioni sulla privacy dei dati, vedi [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog [AWS Shared Responsibility Model and GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).



Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con AWS Mainframe Modernization o altro Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

## Dati raccolti da Mainframe AWS Modernization

AWS Mainframe Modernization raccoglie diversi tipi di dati dall'utente:

- **Application configuration:** Si tratta di un file JSON creato per configurare l'applicazione. Contiene le opzioni disponibili per le diverse opzioni offerte da AWS Mainframe Modernization. Il file contiene anche informazioni su AWS risorse dipendenti come i percorsi di Amazon Simple Storage Service in cui sono archiviati gli artefatti dell'applicazione o l'Amazon Resource Name (ARN) per il luogo in AWS Secrets Manager cui sono archiviate le credenziali del database.
- **Application executable (binary):** Si tratta di un file binario che compili e che intendi distribuire su Mainframe Modernization. AWS
- **Application JCL or scripts:** questo codice sorgente gestisce i processi in batch o altre elaborazioni per conto dell'applicazione.
- **User application data:** Quando importate set di dati, AWS Mainframe Modernization li memorizza nel database relazionale in modo che l'applicazione possa accedervi.
- **Application source code:** Tramite Amazon AppStream 2.0, AWS Mainframe Modernization fornisce un ambiente di sviluppo per scrivere e compilare codice.

AWS Mainframe Modernization archivia questi dati in modo nativo. AWS I dati che raccogliamo da te sono archiviati in un bucket Amazon S3 gestito da AWS Mainframe Modernization. Quando distribuisce un'applicazione, AWS Mainframe Modernization scarica i dati su un'istanza Amazon Elastic Compute Cloud supportata da Amazon Elastic Block Store. Quando viene attivata la pulizia, i dati vengono rimossi dal volume Amazon EBS e da Amazon S3. I volumi Amazon EBS sono single-tenant, il che significa che un'istanza viene utilizzata per un cliente. Le istanze non vengono mai condivise. Quando elimini un ambiente di runtime, viene eliminato anche il volume Amazon EBS. Quando elimini un'applicazione, gli artefatti e la configurazione vengono eliminati da Amazon S3.

I log delle applicazioni sono archiviati in Amazon CloudWatch. Vengono esportati anche i messaggi di registro delle applicazioni dei clienti CloudWatch in. I CloudWatch log possono contenere dati riservati ai clienti, come dati aziendali o informazioni sulla sicurezza (nei messaggi di debug). Per ulteriori informazioni, consulta [Monitoraggio della modernizzazione AWS del mainframe con Amazon CloudWatch](#).

Inoltre, se scegli di collegare uno o più file system Amazon Elastic File System o Amazon FSx al tuo ambiente di runtime, i dati all'interno di tali sistemi verranno archiviati in. AWS Dovrai ripulire quei dati se decidi di smettere di usare i file system.

Puoi utilizzare tutte le opzioni di crittografia di Amazon S3 disponibili per proteggere i tuoi dati quando li inserisci nel bucket Amazon S3 utilizzato da Mainframe Modernization per la distribuzione di applicazioni e AWS l'importazione di set di dati. Inoltre, puoi utilizzare le opzioni di crittografia Amazon EFS e Amazon FSx se colleghi uno o più di questi file system al tuo ambiente di runtime.

## Crittografia dei dati inattiva per il servizio di AWS modernizzazione del mainframe

AWS La modernizzazione del mainframe si integra AWS Key Management Service per fornire una crittografia lato server (SSE) trasparente su tutte le risorse dipendenti che archiviano i dati in modo permanente, vale a dire Amazon Simple Storage Service, Amazon DynamoDB e Amazon Elastic Block Store. AWS Mainframe Modernization crea e gestisce al tuo posto chiavi di crittografia simmetriche. AWS KMS AWS KMS

La crittografia predefinita dei dati a riposo aiuta a ridurre il sovraccarico operativo e la complessità associati alla protezione dei dati sensibili. Allo stesso tempo, consente di migrare applicazioni che richiedono una rigorosa conformità alla crittografia e requisiti normativi.

Non è possibile disabilitare questo livello di crittografia o selezionare un tipo di crittografia alternativo quando si creano ambienti e applicazioni di runtime.

Puoi utilizzare la tua chiave gestita dal cliente per le applicazioni di modernizzazione del AWS mainframe e gli ambienti di runtime per crittografare le risorse Amazon S3 e Amazon EBS.

Per le tue applicazioni di modernizzazione del AWS mainframe, puoi utilizzare questa chiave per crittografare la definizione dell'applicazione e altre risorse applicative, come i file JCL, che vengono salvati nel bucket Amazon S3 creato nell'account del servizio. Per ulteriori informazioni, consulta [Creazione di un'applicazione](#).

Per gli AWS ambienti di runtime di modernizzazione del AWS mainframe, Mainframe Modernization utilizza la chiave gestita dal cliente per crittografare il volume Amazon EBS che crea e collega alla tua istanza AWS Amazon EC2 per la modernizzazione del mainframe, anch'essa presente nell'account del servizio. Per ulteriori informazioni, consulta [Crea un ambiente di runtime](#).

### Note

Le risorse DynamoDB sono sempre crittografate utilizzando Chiave gestita da AWS un account del servizio di modernizzazione AWS del mainframe. Non è possibile crittografare le risorse DynamoDB utilizzando una chiave gestita dal cliente.

AWS Mainframe Modernization utilizza la chiave gestita dal cliente per le seguenti attività:

- Ridistribuzione di un'applicazione.
- Sostituzione di un' AWS istanza Amazon EC2 per la modernizzazione del mainframe.

AWS Mainframe Modernization non utilizza la chiave gestita dal cliente per crittografare i database Amazon Relational Database Service o Amazon Aurora, le code di Amazon Simple Queue Service e le cache ElastiCache Amazon create per supportare AWS un'applicazione di modernizzazione del mainframe, poiché nessuna di esse contiene dati dei clienti.

Per ulteriori informazioni, consulta [Chiavi gestite dal cliente \(CMK\)](#) nella Guida per sviluppatori AWS Key Management Service .

La tabella seguente riassume il modo in cui Mainframe Modernization crittografa i dati sensibili. AWS

Tipo di dati	Chiave gestita da AWS crittografia	crittografia a chiave gestita dal cliente
Definition  Contiene la definizione per una particolare applicazione.	Abilitato	Abilitato
EnvironmentSummary  Contiene informazioni sull'ambiente di runtime.	Abilitato	Abilitato

Tipo di dati	Chiave gestita da AWS crittografia	crittografia a chiave gestita dal cliente
<b>ApplicationSummary</b>  Contiene informazioni sull'applicazione AWS Mainframe Modernization.	Abilitato	Abilitato
<b>DeploymentSummary</b>  Contiene informazioni sulla distribuzione di un'applicazione di modernizzazione del AWS mainframe.	Abilitato	Abilitato

#### Note

AWS La modernizzazione del mainframe abilita automaticamente la crittografia a riposo, proteggendo gratuitamente i dati sensibili. Chiavi gestite da AWS Tuttavia, l'utilizzo di una chiave gestita dal cliente comporta dei costi AWS KMS. Per ulteriori informazioni sui prezzi, consulta [Prezzi di AWS Key Management Service](#).

Per ulteriori informazioni su AWS KMS, vedere AWS Key Management Service.

## In che modo AWS Mainframe Modernization utilizza le sovvenzioni in AWS KMS

AWS La modernizzazione del mainframe richiede una [concessione](#) per utilizzare la chiave gestita dal cliente.

Quando si crea un'applicazione o un ambiente di runtime o si distribuisce un'applicazione in AWS Mainframe Modernization crittografata con una chiave gestita dal cliente, AWS Mainframe Modernization crea una concessione per conto dell'utente inviando una richiesta a [CreateGrant](#) AWS KMS. Le sovvenzioni AWS KMS vengono utilizzate per consentire a AWS Mainframe Modernization di accedere a una chiave KMS in un account cliente.

AWS La modernizzazione del mainframe richiede la concessione dell'utilizzo della chiave gestita dal cliente per le seguenti operazioni interne:

- [DescribeKey](#)Inviare richieste AWS KMS per verificare che l'ID della chiave simmetrica gestita dal cliente immesso durante la creazione di un'applicazione, un ambiente di runtime o l'implementazione di un'applicazione sia valido.
- Invia [GenerateDataKey](#)richieste per AWS KMS crittografare il volume Amazon EBS collegato alle istanze Amazon EC2 che AWS ospitano ambienti di runtime di modernizzazione del mainframe.
- Invia richieste [Decrypt a per AWS KMS decrittografare](#) i contenuti crittografati su Amazon EBS.

AWS Mainframe Modernization utilizza le AWS KMS concessioni per decrittografare i segreti archiviati in Secrets Manager e durante la creazione di un ambiente di runtime, la creazione o la redistribuzione di un'applicazione e la creazione di una distribuzione. Le sovvenzioni create da Mainframe Modernization supportano le seguenti AWS operazioni:

- Crea o aggiorna una concessione per l'ambiente di runtime:
  - Decrypt
  - Crittografia
  - ReEncryptFrom
  - ReEncryptTo
  - GenerateDataKey
  - DescribeKey
  - CreateGrant
- Creare o redistribuire una concessione per un'applicazione:
  - GenerateDataKey
- Crea una concessione per l'implementazione:
  - Decrypt

Puoi revocare l'accesso alla concessione o rimuovere l'accesso del servizio alla chiave gestita dal cliente in qualsiasi momento. In tal caso, AWS Mainframe Modernization non sarà in grado di accedere a nessuno dei dati crittografati dalla chiave gestita dal cliente, il che influirà sulle operazioni che dipendono dai dati. Ad esempio, se AWS Mainframe Modernization tentasse di accedere a una definizione di applicazione crittografata da una chiave gestita dal cliente senza la concessione di tale chiave, l'operazione di creazione dell'applicazione avrebbe esito negativo.

AWS Mainframe Modernization raccoglie le configurazioni delle applicazioni utente (file JSON) e gli artefatti (file binari ed eseguibili). Crea inoltre metadati che tengono traccia delle varie entità utilizzate per il funzionamento di Mainframe Modernization e crea log e metriche. AWS I log e le metriche visibili ai clienti includono:

- CloudWatch registri che riflettono l'applicazione e il motore di runtime ( AWS Blu Age o Micro Focus).
- CloudWatch metriche per i dashboard operativi.

Inoltre, AWS Mainframe Modernization raccoglie dati e metriche di utilizzo per la misurazione, la reportistica delle attività e così via sui servizi. Questi dati non sono visibili ai clienti.

AWS La modernizzazione del mainframe archivia questi dati in luoghi diversi a seconda del tipo di dati. I dati dei clienti che carichi vengono archiviati in un bucket Amazon S3. I dati del servizio sono archiviati sia in Amazon S3 che in DynamoDB. Quando distribuisce un'applicazione, sia i dati che i dati di servizio vengono scaricati su volumi Amazon EBS. Se scegli di collegare lo storage Amazon EFS o Amazon FSx al tuo ambiente di runtime, i dati archiviati in tali file system vengono scaricati anche nel volume Amazon EBS.

La crittografia a riposo è configurata per impostazione predefinita. Non puoi disabilitarlo o modificarlo. Al momento non è nemmeno possibile modificarne la configurazione.

## Crea una chiave gestita dal cliente

È possibile creare una chiave simmetrica gestita dal cliente utilizzando AWS Management Console o le AWS KMS API.

Per creare una chiave simmetrica gestita dal cliente

Segui la procedura riportata in [Creazione di una chiave simmetrica gestita dal cliente](#) nella Guida per gli sviluppatori di AWS Key Management Service .

### Policy delle chiavi

Le policy chiave controllano l'accesso alla chiave gestita dal cliente. Ogni chiave gestita dal cliente deve avere esattamente una policy delle chiavi, che contiene istruzioni che determinano chi può usare la chiave e come la possono usare. Quando crei la chiave gestita dal cliente, puoi specificare una policy chiave. Per ulteriori informazioni, consulta [Gestione dell'accesso alle chiavi gestite dal cliente](#) nella Guida per gli sviluppatori di AWS Key Management Service .

Per utilizzare la chiave gestita dal cliente con le risorse di modernizzazione del AWS mainframe, nella policy chiave devono essere consentite le seguenti operazioni API:

- [kms:CreateGrant](#): aggiunge una concessione a una chiave gestita dal cliente. Concede l'accesso di controllo a una chiave KMS specificata, che consente l'accesso alle [operazioni AWS di concessione](#) richieste da Mainframe Modernization. Per ulteriori informazioni sull'[utilizzo di Grants](#), [consulta](#) la Guida per gli sviluppatori.AWS Key Management Service

Ciò consente alla modernizzazione AWS del mainframe di eseguire le seguenti operazioni:

- Chiama `GenerateDataKey` per generare una chiave dati crittografata e archivarla, poiché la chiave dati non viene utilizzata immediatamente per crittografare.
- Chiama `Decrypt` per utilizzare la chiave dati crittografata memorizzata per accedere ai dati crittografati.
- Imposta un preside in pensione per consentire al servizio di farlo. `RetireGrant`
- [kms:DescribeKey](#)— Fornisce i dettagli chiave gestiti dal cliente per consentire alla modernizzazione del AWS mainframe di convalidare la chiave.

AWS I requisiti `kms:CreateGrant` e le `kms:DescribeKey` autorizzazioni per la modernizzazione del mainframe sono inclusi nella politica chiave del cliente. AWS Mainframe Modernization utilizza questa politica per creare una sovvenzione a sé stante.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::AccountId:role/ExampleRole"
    },
    "Action": [
      "kms:CreateGrant",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }]
}
```

**Note**

Il ruolo illustrato `Principal` nell'esempio precedente è quello utilizzato per le operazioni di modernizzazione del AWS mainframe come `e. CreateApplication` `CreateEnvironment`

Per ulteriori informazioni su come [specificare le autorizzazioni in una policy](#), consulta la Guida per gli sviluppatori di AWS Key Management Service .

Per informazioni sulla [Risoluzione dei problemi delle chiavi di accesso](#) consulta la Guida per gli sviluppatori di AWS Key Management Service .

## Specificazione di una chiave gestita dal cliente per la modernizzazione del mainframe AWS

È possibile specificare una chiave gestita dal cliente per le seguenti risorse:

- Applicazione
- Ambiente

Quando si crea una risorsa, è possibile specificare la chiave inserendo un ID KMS, che AWS Mainframe Modernization utilizza per crittografare i dati sensibili archiviati dalla risorsa.

- ID KMS: un [identificatore chiave per una chiave gestita](#) dal cliente. Inserisci l'ID della chiave, l'ARN della chiave, il nome dell'alias o l'ARN dell'alias.

È possibile specificare una chiave gestita dal cliente utilizzando AWS Management Console o il AWS CLI

Per specificare la chiave gestita dal cliente durante la creazione di un ambiente di runtime in AWS Management Console, vedere [Crea un ambiente di runtime per la modernizzazione del AWS mainframe](#). Per specificare la chiave gestita dal cliente durante la creazione di un'applicazione in AWS Management Console, vedere [Create un'applicazione di modernizzazione del mainframe AWS](#).

Per aggiungere la chiave gestita dal cliente quando si crea un ambiente di runtime con AWS CLI, specificare il `kms-key-id` parametro nel modo seguente:



```
aws m2 create-environment --engine-type microfocus --instance-type M2.m5.large
--publicly-accessible --engine-version 7.0.3 --name test
--high-availability-config desiredCapacity=2
--kms-key-id myEnvironmentKey
```

Per aggiungere la chiave gestita dal cliente quando si crea un'applicazione con AWS CLI, specificare il `kms-key-id` parametro nel modo seguente:

```
aws m2 create-application --name test-application --description my description
--engine-type microfocus
--definition content="$(jq -c . raw-template.json | jq -R)"
--kms-key-id myApplicationKey
```

## AWS Modernizzazione del mainframe: 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 associa il contesto di crittografia ai dati crittografati. Per decrittografare i dati, nella richiesta deve essere incluso lo stesso contesto di crittografia.

### AWS Modernizzazione del mainframe: contesto di crittografia

AWS La modernizzazione del mainframe utilizza lo stesso contesto di crittografia in tutte le operazioni AWS KMS crittografiche relative a un'applicazione (creazione dell'applicazione e creazione della distribuzione), in cui la chiave è `aws:m2:app` e il valore è l'identificatore univoco dell'applicazione.

### Example

```
"encryptionContextSubset": {
  "aws:m2:app": "a1bc2defabc3defabc4defabcd"
}
```

### Utilizzo del contesto di crittografia per il monitoraggio

Quando si utilizza una chiave simmetrica gestita dal cliente per crittografare le applicazioni o gli ambienti di runtime, è inoltre possibile utilizzare il contesto di crittografia nei record e nei log di controllo per identificare come viene utilizzata la chiave gestita dal cliente.

## Utilizzo del contesto di crittografia per controllare l'accesso alla chiave gestita dal cliente

È possibile utilizzare il contesto di crittografia nelle policy delle chiavi e nelle policy IAM come `conditions` per controllare l'accesso alla chiave simmetrica gestita dal cliente. È possibile utilizzare i vincoli del contesto di crittografia in una concessione.

AWS La modernizzazione del mainframe utilizza un vincolo di contesto di crittografia nelle concessioni per controllare l'accesso alla chiave gestita dal cliente nell'account o nella regione. Il vincolo della concessione richiede che le operazioni consentite dalla concessione utilizzino il contesto di crittografia specificato. L'esempio seguente è una concessione che AWS Mainframe Modernization sfrutta per crittografare gli elementi dell'applicazione durante la creazione di un'applicazione.

```
//This grant is retired immediately after create application finish
{
  "grantee-principal": m2.us-west-2.amazonaws.com,
  "retiring-principal": m2.us-west-2.amazonaws.com,
  "operations": [
    "GenerateDataKey"
  ]
  "condition": {
    "encryptionContextSubset": {
      "aws:m2:app": "a1bc2defabc3defabc4defabcd"
    }
  }
}
```

## Monitoraggio delle chiavi di crittografia per la modernizzazione del mainframe AWS

Quando utilizzi una chiave gestita AWS KMS dal cliente con le tue risorse di modernizzazione del AWS mainframe, puoi utilizzare [AWS CloudTrailAmazon CloudWatch Logs](#) per tenere traccia delle richieste a cui AWS Mainframe Modernization invia. AWS KMS

### Esempi di ambienti di runtime

Gli esempi seguenti sono AWS CloudTrail eventi per `DescribeKey`, `CreateGrantGenerateDataKey`, e per `Decrypt` monitorare le operazioni KMS richiamate da AWS Mainframe Modernization per accedere ai dati crittografati dalla chiave gestita dal cliente:

## DescribeKey

AWS Mainframe Modernization utilizza l'DescribeKey operazione per verificare se la chiave gestita AWS KMS dal cliente associata all'ambiente di runtime esiste nell'account e nella regione.

L'evento di esempio seguente registra l'operazione DescribeKey:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T19:40:26Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-12-06T20:23:43Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "DescribeKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "205.251.233.182",
  "userAgent": "ExampleDesktop/1.0 (V1; OS)",
  "requestParameters": {
    "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
```

```

    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_256_GCM_SHA384",
    "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
  },
  "sessionCredentialFromConsole": "true"
}

```

## CreateGrant

Quando si utilizza una chiave gestita AWS KMS dal cliente per crittografare l'ambiente di runtime, AWS Mainframe Modernization invia diverse CreateGrant richieste per conto dell'utente per eseguire le operazioni KMS necessarie. Alcune delle sovvenzioni create da AWS Mainframe Modernization vengono ritirate immediatamente dopo l'uso. Le altre vengono ritirate quando si elimina l'ambiente di runtime.

L'evento di esempio seguente registra l>CreateGrant operazione per il ruolo di esecuzione Lambda associato al flusso di lavoro Create Environment.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",

```

```

        "accountId": "111122223333",
        "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
        "creationDate": "2022-12-06T20:11:45Z",
        "mfaAuthenticated": "false"
    }
},
"invokedBy": "m2.us-west-2.amazonaws.com"
},
"eventTime": "2022-12-06T20:23:09Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "m2.us-west-2.amazonaws.com",
"userAgent": "m2.us-west-2.amazonaws.com",
"requestParameters": {
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "operations": [
        "Encrypt",
        "Decrypt",
        "ReEncryptFrom",
        "ReEncryptTo",
        "GenerateDataKey",
        "GenerateDataKey",
        "DescribeKey",
        "CreateGrant"
    ],
    "granteePrincipal": "m2.us-west-2.amazonaws.com",
    "retiringPrincipal": "m2.us-west-2.amazonaws.com"
},
"responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
    {

```

```

        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

L'evento di esempio seguente registra l'CreateGrant operazione per il ruolo collegato al servizio del gruppo Auto Scaling. Il ruolo di esecuzione Lambda associato al flusso di lavoro Create Environment chiama questa CreateGrant operazione. Concede l'autorizzazione al ruolo di esecuzione per creare una sottoconcessione rispetto al ruolo collegato ai servizi del gruppo Auto Scaling.

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ARO3YPCLM65MZFUPM4J0:EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",
    "arn": "arn:aws:sts::111122223333:assumed-role/EnvironmentWorkflow-
alpha-CreateEnvironmentLambdaS-1AU4A8VNQEEKN/EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN",
        "accountId": "111122223333",
        "userName": "EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:22:28Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```

```

    }
  }
},
"eventTime": "2022-12-06T20:23:09Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "54.148.236.160",
"userAgent": "aws-sdk-java/2.18.21 Linux/4.14.255-276-224.499.amzn2.x86_64
OpenJDK_64-Bit_Server_VM/11.0.14.1+10-LTS Java/11.0.14.1 vendor/Amazon.com_Inc. md/
internal exec-env/AWS_Lambda_java11 io/sync http/Apache cfg/retry-mode/legacy",
"requestParameters": {
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
  "operations": [
    "Encrypt",
    "Decrypt",
    "ReEncryptFrom",
    "ReEncryptTo",
    "GenerateDataKey",
    "GenerateDataKey",
    "DescribeKey",
    "CreateGrant"
  ],
  "granteePrincipal": "m2.us-west-2.amazonaws.com",
  "retiringPrincipal": "m2.us-west-2.amazonaws.com"
},
"responseElements": {
  "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": false,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],

```

```

    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
      "tlsVersion": "TLSv1.3",
      "cipherSuite": "TLS_AES_256_GCM_SHA384",
      "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
  }
}

```

## GenerateDataKey

Quando abiliti una chiave gestita AWS KMS dal cliente per la tua risorsa dell'ambiente di runtime, Auto Scaling crea una chiave unica per crittografare il volume Amazon EBS associato all'ambiente di runtime. Invia una `GenerateDataKey` richiesta a AWS KMS cui specifica la chiave gestita dal AWS KMS cliente per la risorsa.

L'evento di esempio seguente registra l'operazione `GenerateDataKey`:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ARO3YPCLM65EEXVIEH7D:AutoScaling",
    "arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForAutoScaling/AutoScaling",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/aws-service-role/autoscaling.amazonaws.com/AWSServiceRoleForAutoScaling",
        "accountId": "111122223333",
        "userName": "AWSServiceRoleForAutoScaling"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:23:16Z",
        "mfaAuthenticated": "false"
      }
    }
  }
}

```



```

    }
  },
  "invokedBy": "autoscaling.amazonaws.com"
},
"eventTime": "2022-12-06T20:23:18Z",
"eventSource": "kms.amazonaws.com",
"eventName": "GenerateDataKey",
"awsRegion": "us-west-2",
"sourceIPAddress": "autoscaling.amazonaws.com",
"userAgent": "autoscaling.amazonaws.com",
"requestParameters": {
  "encryptionContext": {
    "aws:ebs:id": "vol-080f7a32d290807f3"
  },
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
  "numberOfBytes": 64
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}

```

## Decrypt

Quando accedi a un ambiente di runtime crittografato, Amazon EBS chiama l'Decryptoperazione per utilizzare la chiave dati crittografata memorizzata per accedere ai dati crittografati.

L'evento di esempio seguente registra l'operazione Decrypt:

```
{
```

```

"eventVersion": "1.08",
"userIdentity": {
  "type": "AWSService",
  "invokedBy": "ebs.amazonaws.com"
},
"eventTime": "2022-12-06T20:23:22Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "ebs.amazonaws.com",
"userAgent": "ebs.amazonaws.com",
"requestParameters": {
  "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
  "encryptionContext": {
    "aws:ebs:id": "vol-080f7a32d290807f3"
  }
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventCategory": "Management"
}

```

## Esempi di applicazioni

Gli esempi seguenti sono AWS CloudTrail eventi per `CreateGrant` e `GenerateDataKey` per il monitoraggio delle operazioni KMS richiamate da AWS Mainframe Modernization per accedere ai dati crittografati dalla chiave gestita dal cliente:

## CreateGrant

Quando utilizzi una chiave gestita AWS KMS dal cliente per crittografare le risorse dell'applicazione, il ruolo di esecuzione Lambda invia CreateGrant una richiesta per tuo conto per accedere alla chiave KMS nel tuo account. AWS La concessione consente al ruolo di esecuzione Lambda di caricare le risorse delle applicazioni del cliente su Amazon S3 utilizzando la chiave gestita dal cliente. Questa concessione viene ritirata immediatamente dopo la creazione dell'applicazione.

L'evento di esempio seguente registra l'operazione CreateGrant:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T21:51:45Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T22:47:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
  "userAgent": "m2.us-west-2.amazonaws.com",
  "requestParameters": {
```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "constraints": {
      "encryptionContextSubset": {
        "aws:m2:app": "a1bc2defabc3defabc4defabcd"
      }
    },
    "retiringPrincipal": "m2.us-west-2.amazonaws.com",
    "operations": [
      "GenerateDataKey"
    ],
    "granteePrincipal": "m2.us-west-2.amazonaws.com"
  },
  "responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

## GenerateDataKey

Quando abiliti una chiave gestita AWS KMS dal cliente per la tua risorsa applicativa, il ruolo di esecuzione Lambda crea una chiave che utilizza per crittografare e caricare i dati dei clienti su Amazon Simple Storage Service. Il ruolo di esecuzione Lambda invia una `GenerateDataKey` richiesta a AWS KMS che specifica la chiave gestita dal AWS KMS cliente per la risorsa.

L'evento di esempio seguente registra l'operazione GenerateDataKey:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0A3YPCLM65CLCEKKC7Z:ApplicationWorkflow-alpha-CreateApplicationVersion-CstWZUn5R4u6",
    "arn": "arn:aws:sts::111122223333:assumed-role/ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B/ApplicationWorkflow-alpha-CreateApplicationVersion-CstWZUn5R4u6",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:iam::111122223333:role/ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B",
        "accountId": "111122223333",
        "userName": "ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T23:28:32Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T23:29:08Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
  "userAgent": "m2.us-west-2.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {
      "aws:m2:app": "a1bc2defabc3defabc4defabcd",
      "aws:s3:arn": "arn:aws:s3:::supernova-processedtemplate-111122223333-us-west-2/111122223333/a1bc2defabc3defabc4defabcd/1/cics-transaction/ZBNKE35.so"
    },
    "keySpec": "AES_256",
  }
}
```

```

    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "responseElements": null,
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": true,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

## Esempi di implementazioni

Gli esempi seguenti sono AWS CloudTrail eventi per `CreateGrant` e `Decrypt` per il monitoraggio delle operazioni KMS richiamate da AWS Mainframe Modernization per accedere ai dati crittografati dalla chiave gestita dal cliente:

### CreateGrant

Quando si utilizza una chiave gestita AWS KMS dal cliente per crittografare le risorse di implementazione, AWS Mainframe Modernization invia due richieste per conto dell'utente. `CreateGrant` La prima concessione riguarda l'attuale ruolo di esecuzione Lambda da chiamare `ListBatchJobScriptFiles` e viene ritirata immediatamente dopo il completamento della distribuzione. La seconda concessione riguarda il ruolo di istanza delimitata di Amazon EC2, in modo che Amazon EC2 possa scaricare le risorse delle applicazioni dei clienti da Amazon S3. Questa concessione viene ritirata quando l'applicazione viene eliminata dall'ambiente di runtime.

L'evento di esempio seguente registra l'operazione `CreateGrant`:

```

{
  "eventVersion": "1.08",

```

```
"userIdentity": {
  "type": "AssumedRole",
  "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
  "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
  "accountId": "111122223333",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
      "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
      "accountId": "111122223333",
      "userName": "Admin"
    },
    "webIdFederationData": {},
    "attributes": {
      "creationDate": "2022-12-06T21:51:45Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "m2.us-west-2.amazonaws.com"
},
"eventTime": "2022-12-06T23:40:07Z",
"eventSource": "kms.amazonaws.com",
"eventName": "CreateGrant",
"awsRegion": "us-west-2",
"sourceIPAddress": "m2.us-west-2.amazonaws.com",
"userAgent": "m2.us-west-2.amazonaws.com",
"requestParameters": {
  "operations": [
    "Decrypt"
  ],
  "constraints": {
    "encryptionContextSubset": {
      "aws:m2:app": "a1bc2defabc3defabc4defabcd"
    }
  },
  "granteePrincipal": "m2.us-west-2.amazonaws.com",
  "retiringPrincipal": "m2.us-west-2.amazonaws.com",
  "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": {
```

```

    "grantId":
      "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }

```

## Decrypt

Quando accedi a una distribuzione, Amazon EC2 chiama l'Decryptoperazione per utilizzare la chiave dati crittografata archiviata per decrittografare e scaricare i dati crittografati dei clienti da Amazon S3.

L'evento di esempio seguente registra l'operazione Decrypt:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AR0A3YPCLM65BSPZ37E6G:m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "arn": "arn:aws:sts::111122223333:assumed-role/
SupernovaEnvironmentInstanceScopeDownRole/m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",

```



```

        "arn": "arn:aws:iam::111122223333:role/
SupernovaEnvironmentInstanceScopeDownRole",
        "accountId": "111122223333",
        "userName": "SupernovaEnvironmentInstanceScopeDownRole"
    },
    "webIdFederationData": {},
    "attributes": {
        "creationDate": "2022-12-06T23:19:29Z",
        "mfaAuthenticated": "false"
    }
},
"invokedBy": "m2.us-west-2.amazonaws.com"
},
"eventTime": "2022-12-06T23:40:15Z",
"eventSource": "kms.amazonaws.com",
"eventName": "Decrypt",
"awsRegion": "us-west-2",
"sourceIPAddress": "m2.us-west-2.amazonaws.com",
"userAgent": "m2.us-west-2.amazonaws.com",
"requestParameters": {
    "encryptionContext": {
        "aws:m2:app": "a1bc2defabc3defabc4defabcdm",
        "aws:s3:arn": "arn:aws:s3:::supernova-processedtemplate-111122223333-us-
west-2/111122223333/a1bc2defabc3defabc4defabcdm/1/cics-transaction/BBANK40P.so"
    },
    "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"readOnly": true,
"resources": [
    {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"

```

}

## Ulteriori informazioni

Le seguenti risorse forniscono ulteriori informazioni sulla crittografia dei dati a riposo.

- Per ulteriori informazioni su [Concetti base di AWS Key Management Service](#), consulta la Guida per gli sviluppatori di AWS Key Management Service .
- Per ulteriori informazioni su [Best Practice di sicurezza per AWS Key Management Service](#) nella Guida per sviluppatori di AWS Key Management Service .

## Crittografia in transito

Per le applicazioni interattive che fanno parte di carichi di lavoro transazionali, gli scambi di dati tra l'emulatore di terminale e l'endpoint del servizio di modernizzazione del AWS mainframe per il protocollo TN3270 non sono crittografati in transito. Se l'applicazione richiede la crittografia in transito, potresti voler implementare alcuni meccanismi di tunneling aggiuntivi.

AWS La modernizzazione del mainframe utilizza HTTPS per crittografare le API del servizio. Tutte le altre comunicazioni all'interno di AWS Mainframe Modernization sono protette dal servizio VPC o dal gruppo di sicurezza, oltre che da HTTPS. AWS Mainframe Modernization trasferisce gli artefatti, le configurazioni e i dati delle applicazioni. Gli artefatti dell'applicazione vengono copiati da un bucket Amazon S3 di tua proprietà, così come i dati dell'applicazione. Puoi fornire configurazioni delle applicazioni utilizzando un collegamento ad Amazon S3 o caricando un file localmente.

La crittografia di base in transito è configurata di default, ma non si applica al protocollo TN3270. AWS La modernizzazione del mainframe utilizza HTTPS per gli endpoint API, anch'essi configurati di default.

## Identity and Access Management per la modernizzazione del AWS mainframe

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare AWS le risorse di modernizzazione del mainframe. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

## Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come funziona la modernizzazione AWS del mainframe con IAM](#)
- [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)
- [Risoluzione dei problemi relativi alla modernizzazione del AWS mainframe, all'identità e all'accesso](#)
- [Utilizzo di ruoli collegati ai servizi per la modernizzazione del mainframe](#)

## Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto nella modernizzazione del AWS mainframe.

**Utente del servizio:** se utilizzi il servizio di modernizzazione del AWS mainframe per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più funzionalità di modernizzazione AWS del mainframe per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS Mainframe Modernization, consulta [Risoluzione dei problemi relativi alla modernizzazione del AWS mainframe, all'identità e all'accesso](#)

**Amministratore del servizio:** se sei responsabile delle risorse per la modernizzazione del AWS mainframe della tua azienda, probabilmente hai pieno accesso alla modernizzazione del mainframe. AWS Il tuo compito è determinare a quali funzionalità e risorse di modernizzazione AWS del mainframe devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con la modernizzazione del AWS mainframe, consulta [Come funziona la modernizzazione AWS del mainframe con IAM](#)

**Amministratore IAM:** se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere policy per gestire l'accesso alla modernizzazione del AWS mainframe. Per visualizzare esempi di policy basate sull'identità di modernizzazione del AWS mainframe che puoi utilizzare in IAM, consulta [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)

## Autenticazione con identità

L'autenticazione è il modo in cui accedi utilizzando le tue credenziali di identità. AWS Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente di IAM.

### Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente di IAM.

## Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, un provider di identità Web AWS Directory Service, la directory Identity Center o qualsiasi utente che accede Servizi AWS utilizzando credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta [Cos'è Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente di IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente di IAM.

## Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per ulteriori informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - **Sessioni di accesso diretto (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per

effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) assunto da un servizio per eseguire operazioni per conto dell'utente. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS CLI è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente di IAM.

## Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente di IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. Successivamente l'amministratore può aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall' AWS CLI o dall' AWS API.

## Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente di IAM.

## Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.



## Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

## Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente di IAM.

## Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

## Come funziona la modernizzazione AWS del mainframe con IAM

Prima di utilizzare IAM per gestire l'accesso alla modernizzazione del AWS mainframe, scopri quali funzionalità IAM sono disponibili per l'uso con la modernizzazione del mainframe. AWS

### Funzionalità IAM che puoi utilizzare con Mainframe Modernization AWS

Funzionalità IAM	AWS Supporto per la modernizzazione del mainframe
<a href="#">Policy basate su identità</a>	Sì
<a href="#">Policy basate su risorse</a>	No
<a href="#">Azioni di policy</a>	Sì
<a href="#">Risorse relative alle policy</a>	Sì
<a href="#">Chiavi di condizione delle policy</a>	Sì
<a href="#">Liste di controllo degli accessi (ACL)</a>	No
<a href="#">ABAC (tag nelle policy)</a>	Sì
<a href="#">Credenziali temporanee</a>	Sì
<a href="#">Inoltro delle sessioni di accesso (FAS)</a>	Sì
<a href="#">Ruoli di servizio</a>	Sì
<a href="#">Ruoli collegati al servizio</a>	Sì

Per avere una visione di alto livello di come la modernizzazione del AWS mainframe e gli altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

## Politiche basate sull'identità per la modernizzazione del mainframe AWS

Supporta le policy basate su identità Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

## Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS

Per visualizzare esempi di politiche basate sull'identità di modernizzazione del AWS mainframe, vedere. [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)

## Politiche basate sulle risorse nell'ambito della modernizzazione del mainframe AWS

Supporta le policy basate su risorse No

Le policy basate su risorse sono documenti di policy JSON che è possibile allegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è allegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

## Azioni politiche per la modernizzazione del AWS mainframe

Supporta le azioni di policy

Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a una policy. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni di modernizzazione del AWS mainframe, consulta [Azioni definite dalla modernizzazione del AWS mainframe nel Service Authorization Reference](#).

Le azioni politiche in AWS Mainframe Modernization utilizzano il seguente prefisso prima dell'azione:

```
m2
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
  "m2:StartApplication",  
  "m2:StopApplication"
```

```
]
```

È possibile specificare più operazioni tramite caratteri jolly (\*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola `List`, includi la seguente operazione:

```
"Action": "m2:List*"
```

Per visualizzare esempi di politiche basate sull'identità di modernizzazione del AWS mainframe, vedere [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)

## Risorse politiche per la modernizzazione del mainframe AWS

Supporta le risorse di policy

Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

È possibile limitare l'accesso a risorse specifiche di modernizzazione del AWS mainframe utilizzando i relativi ARN per identificare la risorsa a cui si applica la policy IAM. Per ulteriori informazioni sul formato degli ARN, consulta [Amazon Resource Names \(ARN\)](#) nella Riferimenti generali di AWS.

Ad esempio, un ambiente di modernizzazione AWS del mainframe ha il seguente ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:env/service-generated-unique-identifier"
```

Un'applicazione di modernizzazione AWS del mainframe ha il seguente ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:app/service-generated-unique-identifier"
```

Non tutte le azioni di modernizzazione del AWS mainframe supportano le autorizzazioni a livello di risorsa. Per le azioni che non supportano le autorizzazioni a livello di risorsa, è necessario utilizzare il carattere jolly (\*).

Le seguenti azioni di modernizzazione del AWS mainframe non supportano le autorizzazioni a livello di risorsa.

```
ListApplications
    ListApplicationVersions
    ListBatchJobDefinitions
    ListBatchJobExecutions
    ListDataSetImportHistory
    ListDataSets
    ListDeployments
    ListEngineVersions
    ListEnvironments
    ListTagsForResource
```

Per visualizzare un elenco dei tipi di risorse per la modernizzazione del AWS mainframe e i relativi ARN, consulta [Resources](#) Defined by Mainframe Modernization nel Service Authorization Reference. AWS Per sapere con quali azioni è possibile specificare l'ARN di ogni risorsa, vedere [Azioni definite dalla modernizzazione del AWS mainframe](#).

Per visualizzare esempi di politiche basate sull'identità di modernizzazione del AWS mainframe, vedere. [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)

## AWS Autorizzazioni dell'API di modernizzazione del mainframe: riferimento ad azioni, risorse e condizioni

Quando scrivi policy di autorizzazione da allegare a un'identità IAM (policy basate sull'identità), puoi utilizzare la seguente tabella come riferimento. La tabella include quanto segue:

- Ogni operazione dell'API di modernizzazione del AWS mainframe
- Le operazioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'operazione

- La AWS risorsa per la quale è possibile concedere le autorizzazioni

Specifica le operazioni nel campo `Action` della policy e il valore della risorsa nel campo `Resource` della policy.

È possibile utilizzare le chiavi di condizione AWS globali nelle politiche di modernizzazione del AWS mainframe per esprimere le condizioni. Per un elenco completo delle AWS chiavi, consulta [Available Global Condition Keys](#) nella IAM User Guide.

#### Note

Per specificare un'operazione, utilizza il prefisso `m2` : seguito dal nome dell'operazione API (ad esempio, `m2:CreateApplication`).

AWS API di modernizzazione del mainframe e autorizzazioni richieste per le azioni

AWS Operazioni delle API di modernizzazione del mainframe	Autorizzazioni necessarie (operazioni API)	Risorse
<a href="#">CancelBatchJobExecution</a>		Applicazione
<a href="#">CreateApplication</a>	<code>s3:GetObject</code> <code>s3:ListBucket</code>	Applicazione
<a href="#">CreateDataSetImportTask</a>	<code>m2:CreateDataSetImportTask</code> <code>s3:GetObject</code>	Applicazione
<a href="#">CreateDeployment</a>	<code>elasticloadbalancing:AddTags</code> <code>elasticloadbalancing:CreateListener</code> <code>elasticloadbalancing:CreateTargetGroup</code>	Applicazione

AWS Operazioni delle API di modernizzazione del mainframe	Autorizzazioni necessarie (operazioni API)	Risorse
	elasticloadbalancing:RegisterTargets	
<a href="#">CreateEnvironment</a>	ec2:CreateNetworkInterface ec2:CreateNetworkInterfacePermission ec2:DescribeNetworkInterfaces ec2:DescribeSecurityGroups ec2:DescribeSubnets ec2:DescribeVpcAttribute ec2:DescribeVpcs ec2:ModifyNetworkInterfaceAttribute elasticfilesystem:DescribeMountTargets elasticloadbalancing:AddTags elasticloadbalancing:CreateLoadBalancer fsx:DescribeFileSystems iam:CreateServiceLinkedRole	Ambiente



AWS Operazioni delle API di modernizzazione del mainframe	Autorizzazioni necessarie (operazioni API)	Risorse
<a href="#">DeleteApplication</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup  logs:DeleteLogDelivery	Applicazione
<a href="#">DeleteApplicationFromEnvironment</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup	Applicazione  Ambiente
<a href="#">DeleteEnvironment</a>	elasticloadbalancing:DeleteLoadBalancer	Ambiente
<a href="#">GetApplication</a>		Applicazione
<a href="#">GetApplicationVersion</a>		Applicazione
<a href="#">GetBatchJobExecution</a>		Applicazione
<a href="#">GetDataSetDetails</a>		Applicazione
<a href="#">GetDataSetImportTask</a>		Applicazione
<a href="#">GetDeployment</a>		Applicazione
<a href="#">GetEnvironment</a>		Ambiente
<a href="#">ListApplications</a>		*
<a href="#">ListApplicationVersions</a>		*

AWS Operazioni delle API di modernizzazione del mainframe	Autorizzazioni necessarie (operazioni API)	Risorse
<a href="#">ListBatchJobDefinitions</a>		*
<a href="#">ListBatchJobExecutions</a>		*
<a href="#">ListDataSetImportHistory</a>		*
<a href="#">ListDataSets</a>		*
<a href="#">ListDeployments</a>		*
<a href="#">ListEngineVersions</a>		*
<a href="#">ListEnvironments</a>		*
<a href="#">ListTagsForResource</a>		*
<a href="#">StartApplication</a>		Applicazione
<a href="#">StartBatchJob</a>		Applicazione
<a href="#">StopApplication</a>		Applicazione
<a href="#">TagResource</a>		*
<a href="#">UntagResource</a>		*
<a href="#">UpdateApplication</a>	s3:GetObject s3:ListBucket	Applicazione
<a href="#">UpdateEnvironment</a>		Ambiente

## Le condizioni fondamentali delle policy per AWS la modernizzazione del mainframe

Supporta le chiavi di condizione delle policy specifiche del servizio	Si
---	----

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Le seguenti chiavi di condizione sono specifiche della modernizzazione del AWS mainframe

```
m2:EngineType
    m2:InstanceType
```

Per visualizzare un elenco delle chiavi delle condizioni di modernizzazione del AWS mainframe, consulta [Condition Keys for AWS Mainframe Modernization nel Service Authorization Reference](#). Per sapere con quali azioni e risorse è possibile utilizzare una chiave di condizione, consulta [Actions Defined by Mainframe Modernization](#). AWS

Per visualizzare esempi di politiche basate sull'identità di modernizzazione del AWS mainframe, vedere. [Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS](#)

## Liste di controllo degli accessi (ACL) nella modernizzazione del mainframe AWS

Supporta le ACL

No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

## Controllo degli accessi basato sugli attributi (ABAC) con modernizzazione del mainframe AWS

Supporta ABAC (tag nelle policy)

Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In, questi attributi sono chiamati AWS tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta [Che cos'è ABAC?](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

## Utilizzo di credenziali temporanee con AWS Mainframe Modernization

Supporta le credenziali temporanee Sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla [Servizi AWS compatibilità con IAM nella IAM User Guide](#).

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Cambio di un ruolo \(console\)](#) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

## Sessioni di accesso diretto per AWS la modernizzazione del mainframe

Supporta sessioni di accesso diretto (FAS) Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le operazioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

**⚠ Important**

Questi token consentono a AWS Mainframe Modernization di accedere ai dati dei clienti senza il tuo consenso esplicito; ad esempio, AWS Mainframe Modernization distribuisce artefatti applicativi con dati aziendali associati da un bucket Amazon S3 senza ottenere l'autorizzazione esplicita dal cliente. Potrebbe essere necessario aggiornare di conseguenza qualsiasi documentazione di conformità.

## Ruoli di servizio per la AWS modernizzazione del mainframe

Supporta i ruoli di servizio	Sì
------------------------------	----

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente di IAM.

AWS La modernizzazione del mainframe supporta i ruoli di servizio per le fasi di attività (transazioni, lavori, fine settimana o completamento, ecc.).

**⚠ Warning**

La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di modernizzazione del mainframe. AWS Modificate i ruoli di servizio solo quando AWS Mainframe Modernization fornisce indicazioni in tal senso.

## Scelta di un ruolo IAM nella modernizzazione del mainframe AWS

Se in precedenza hai creato un ruolo IAM che le tue applicazioni in esecuzione su Amazon EC2 possono assumere, puoi scegliere questo ruolo quando crei un modello di lancio o una configurazione di avvio. AWS Mainframe Modernization ti fornisce un elenco di ruoli tra cui scegliere. Durante la creazione di questi ruoli, è importante associare le policy IAM con privilegi minimi per limitare l'accesso alle chiamate API specifiche richieste dall'applicazione. Per ulteriori informazioni, consulta [Ruolo IAM per le applicazioni in esecuzione sulle istanze Amazon EC2](#) nella Guida per l'utente di Amazon EC2 Auto Scaling.

## Ruoli collegati ai servizi per la modernizzazione del mainframe AWS

Supporta i ruoli collegati ai servizi

Sì

Un ruolo collegato ai servizi è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per informazioni dettagliate sulla creazione o la gestione dei ruoli collegati ai servizi di AWS Mainframe Modernization, consulta [Utilizzo di ruoli collegati ai servizi per la modernizzazione del mainframe](#)

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta [Servizi AWS supportati da IAM](#). Trova un servizio nella tabella che include un Yes nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

## Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare le risorse di modernizzazione del mainframe. AWS. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o AWS I'API. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS Mainframe Modernization, incluso il formato degli ARN per ciascun tipo di risorsa, consulta [Actions, Resources and Condition Keys for AWS Mainframe Modernization nel Service Authorization Reference](#).

### Argomenti

- [Best practice per le policy](#)

- [Utilizzo della console di modernizzazione del mainframe AWS](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)

## Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse di modernizzazione del mainframe nel tuo account. AWS Queste operazioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le policy gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori



informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

## Utilizzo della console di modernizzazione del mainframe AWS

Per accedere alla console di modernizzazione del AWS mainframe, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirvi di elencare e visualizzare i dettagli sulle risorse di modernizzazione del AWS mainframe presenti nel vostro Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso o l' AWS CLI API. Al contrario, concedi l'accesso solo alle operazioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano continuare a utilizzare la console di modernizzazione del AWS mainframe, collega anche la modernizzazione del AWS mainframe ConsoleAccess o la policy ReadOnly AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

## Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono allegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
```

```

        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

## Risoluzione dei problemi relativi alla modernizzazione del AWS mainframe, all'identità e all'accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS Mainframe Modernization e IAM.

### Argomenti

- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse di modernizzazione AWS del mainframe](#)

### Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'iam:PassRoleazione, le tue policy devono essere aggiornate per consentirti di assegnare un ruolo a AWS Mainframe Modernization.

Alcuni Servizi AWS consentono di trasferire un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in AWS Mainframe Modernization. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse di modernizzazione AWS del mainframe

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Mainframe Modernization supporta queste funzionalità, consulta [Come funziona la modernizzazione AWS del mainframe con IAM](#)
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM di un altro Account AWS utente di tua proprietà nella](#) IAM User Guide. Account AWS
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente di IAM.

- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente di IAM.

## Utilizzo di ruoli collegati ai servizi per la modernizzazione del mainframe

AWS Mainframe Modernization utilizza [ruoli collegati al servizio AWS Identity and Access Management\(IAM\)](#). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente alla modernizzazione del mainframe. I ruoli collegati ai servizi sono predefiniti da Mainframe Modernization e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per conto dell'utente. AWS

Un ruolo collegato ai servizi semplifica la configurazione di Mainframe Modernization perché non è necessario aggiungere manualmente le autorizzazioni necessarie. Mainframe Modernization definisce le autorizzazioni dei ruoli collegati ai servizi e, salvo diversa definizione, solo Mainframe Modernization può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Ciò protegge le risorse di modernizzazione del mainframe perché non è possibile rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consulta [Servizi AWS che funzionano con IAM](#) e cerca i servizi che riportano Yes (Sì) nella colonna Service-linked roles (Ruoli collegati ai servizi). Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

## Autorizzazioni di ruolo collegate ai servizi per la modernizzazione del mainframe

La modernizzazione del mainframe utilizza il ruolo collegato ai servizi denominato `AWSServiceRoleForAWSM2`: configura la rete per connettersi al tuo VPC e accedere a risorse come i file system.

Il ruolo `AWSServiceRoleForAWSM 2` collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `m2.amazonaws.com`

La politica di autorizzazione dei ruoli denominata `AWSM2ServicePolicy` consente a Mainframe Modernization di completare le seguenti azioni sulle risorse specificate:

- Crea, elimina, descrivi e collega le autorizzazioni alle interfacce di rete Amazon EC2 per l'ambiente di modernizzazione del mainframe per stabilire la connettività al VPC del cliente.
- Registra o annulla la registrazione delle voci da Elastic Load Balancing, che è il modo in cui i clienti si connettono all'ambiente di modernizzazione del mainframe.
- Descrivi il file system Amazon EFS o Amazon FSx, se utilizzato.
- Invia i parametri al cliente CloudWatch dall'ambiente di runtime.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2>DeleteNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:CreateNetworkInterfacePermission",
        "ec2:ModifyNetworkInterfaceAttribute"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:DescribeMountTargets"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "elasticloadbalancing:RegisterTargets",
        "elasticloadbalancing:DeregisterTargets"
      ],
      "Resource": "*"
    },
    {
```

```
"Effect": "Allow",
"Action": [
  "fsx:DescribeFileSystems"
],
"Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:PutMetricData"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "cloudwatch:namespace": [
        "AWS/M2"
      ]
    }
  }
}
]
```

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Creazione di un ruolo legato ai servizi per la modernizzazione del mainframe

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando create un ambiente di runtime nell'API, nella o nell'AWSAPIAWS Management Console, Mainframe Modernization crea automaticamente il ruolo collegato ai servizi. AWS CLI

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un ambiente di runtime, Mainframe Modernization crea nuovamente il ruolo collegato ai servizi per te.

## Modifica di un ruolo collegato ai servizi per la modernizzazione del mainframe

La modernizzazione del mainframe non consente di modificare i 2 ruoli collegati ai servizi. **AWSServiceRoleForAWSM** Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione

utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Eliminazione di un ruolo collegato ai servizi per la modernizzazione del mainframe

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

### Note

Se il servizio di modernizzazione del mainframe utilizza il ruolo quando si tenta di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse di modernizzazione del mainframe utilizzate dal 2 AWSServiceRoleForAWSM

- Eliminare gli ambienti di runtime in Mainframe Modernization. Assicurati di eliminare le applicazioni da un ambiente prima di eliminare l'ambiente stesso.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAMAWS CLI, o l'AWSAPI per eliminare i AWSServiceRoleForAWSM 2 ruoli collegati ai servizi. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

## Regioni supportate per i ruoli collegati ai servizi di modernizzazione del mainframe

La modernizzazione del mainframe supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [Regioni ed endpoint di AWS](#).

## Convalida della conformità per la conformità per la conformitàAWS per valida della conformità per la conformità

Revisori di terze parti valutano la sicurezza e la conformità per la sicurezza e la conformitàAWS della conformità per la conformità della conformità per la sicurezza e la conformità di della conformità per laAWS conformità della conformità di Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco dei servizi AWS coperti da programmi di conformità specifici, consulta [Servizi AWS coperti dal programma di compliance](#). Per informazioni generali, consulta [Programmi per la conformità di AWS](#).

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La tua responsabilità per la conformità dell'utilizzoAWS di Amazon è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e normative applicabili. AWSfornisce le seguenti risorse per facilitare la conformità:

- [Security and Compliance Quick Start Guides](#) (Guide Quick Start Sicurezza e compliance) (Guide Quick Start Sicurezza e compliance): queste guide alla distribuzione illustrano considerazioni relative all'architettura e forniscono procedure per la distribuzione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- [Whitepaper sulla progettazione per la sicurezza HIPAA e la conformità](#): questo whitepaper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni conformi ai requisiti HIPAA.
- [Risorse per la conformità AWS](#): una raccolta di cartelle di lavoro e guide suddivise per settore e area geografica.
- [Valutazione delle risorse con le regole](#) nella Guida per gli sviluppatori di AWS Config: AWS Config valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti industriali.
- [AWS Security Hub](#): Questo servizio AWS fornisce una visione completa dello stato di sicurezza all'interno di AWS che consente di verificare la conformità con gli standard e le best practice di sicurezza del settore.

## Resilienza inAWSMainframe Modernization

L'infrastruttura globale di AWS è basata su regioni AWS e zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le Zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili, rispetto alle infrastrutture a data center singolo o multiplo.

Per ulteriori informazioni sulle regioni AWS e sulle zone di disponibilità, consulta [Infrastruttura globale di AWS](#).



## Sicurezza dell'infrastruttura in AWS Mainframe Modernization

Come servizio gestito, AWS Mainframe Modernization è protetto dalla sicurezza di rete globale AWS. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina [Sicurezza del cloud AWS](#). Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina [Protezione dell'infrastruttura](#) nel Pilastro della sicurezza di AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere a Mainframe Modernization attraverso la rete. I clienti devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

## Accesso AWS Mainframe Modernization tramite un endpoint di interfaccia () AWS PrivateLink

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e AWS Mainframe Modernization. Puoi accedere a Mainframe Modernization 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 accedere alla modernizzazione del mainframe.

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creato un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Si tratta di interfacce di rete gestite dai richiedenti che fungono da punto di ingresso per il traffico destinato alla modernizzazione del mainframe.

Per ulteriori informazioni, consulta la sezione [Accesso a Servizi AWS tramite AWS PrivateLink](#) nella Guida di AWS PrivateLink.

## Considerazioni sulla modernizzazione del mainframe

[Prima di configurare un endpoint di interfaccia per la modernizzazione del mainframe, consulta le considerazioni nella guida. AWS PrivateLink](#)

Mainframe Modernization supporta l'esecuzione di chiamate a tutte le sue azioni API tramite l'endpoint dell'interfaccia.

## Crea un endpoint di interfaccia per la modernizzazione del mainframe

Puoi creare un endpoint di interfaccia per la modernizzazione del mainframe utilizzando 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 la modernizzazione del mainframe utilizzando il seguente nome di servizio:

```
com.amazonaws.region.m2
```

Se si abilita il DNS privato per l'endpoint di interfaccia, è possibile effettuare richieste API a Mainframe Modernization utilizzando il nome DNS regionale predefinito. Ad esempio, `m2.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 alla modernizzazione del mainframe tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito alla modernizzazione del mainframe dal tuo VPC, collega una policy personalizzata per gli endpoint all'endpoint dell'interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (utenti e ruoli Account AWS 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 sugli endpoint VPC per le azioni di modernizzazione del mainframe

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Quando alleggi questa policy all'endpoint dell'interfaccia, concede l'accesso alle azioni di modernizzazione del mainframe elencate per tutti i principali utenti su tutte le risorse.

```
//Example of an endpoint policy where access is granted to the
//listed AWS Mainframe Modernization actions for all principals on all resources
{"Statement": [
  {"Principal": "*",
    "Effect": "Allow",
    "Action": [
      "m2:ListApplications",
      "m2:ListEnvironments",
      "m2:ListDeployments"
    ],
    "Resource": "*"
  }
]
```

```
//Example of an endpoint policy where access is denied to all the
//AWS Mainframe Modernization CREATE actions for all principals on all resources
{"Statement": [
  {"Principal": "*",
    "Effect": "Deny",
    "Action": [
      "m2:Create*"
    ],
    "Resource": "*"
  }
]
```

# Monitoraggio della AWS modernizzazione del mainframe

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni della modernizzazione del AWS mainframe e delle altre soluzioni AWS. AWS fornisce i seguenti strumenti di monitoraggio per monitorare la modernizzazione del AWS mainframe, segnalare quando qualcosa non va e intraprendere azioni automatiche quando necessario:

- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi tenere CloudWatch traccia dell'utilizzo della CPU o di altri parametri delle tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- 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](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account hanno chiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la [AWS CloudTrail Guida per l'utente](#).

## Monitoraggio della modernizzazione AWS del mainframe con Amazon CloudWatch

È possibile monitorare la modernizzazione del AWS mainframe utilizzando CloudWatch, che raccoglie dati grezzi e li elabora in metriche leggibili e quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web. È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

Le tabelle seguenti elencano le metriche e le dimensioni per la modernizzazione del AWS mainframe. Il namespace per queste metriche è. AWS/M2

## Metriche dell'ambiente di runtime

Parametro	Descrizione
CPUUtilization	L'utilizzo della CPU delle istanze nell'ambiente.  Dimensione: EnvironmentID  Unità: percentuale  Statistiche valide: Average (Media), Minimum, Maximum
InboundNetworkThroughput	Throughput di rete in entrata delle istanze presenti nell'ambiente.  Dimensione: EnvironmentID  Unità: byte al secondo  Statistiche valide: Average (Media), Minimum, Maximum
MemoryUtilization	L'utilizzo della memoria delle istanze nell'ambiente.  Dimensione: EnvironmentID  Unità: percentuale  Statistiche valide: Average (Media), Minimum, Maximum
OutboundNetworkThroughput	Throughput di rete in uscita delle istanze nell'ambiente.  Dimensione: EnvironmentID

Parametro	Descrizione
	Unità: byte al secondo  Statistiche valide: Average (Media), Minimum, Maximum

## Parametri di applicazioni

Parametro	Descrizione
BatchJobCompletedCount	<p>Il numero di lavori completati durante l'intervallo di tempo.</p> <p>Questa metrica è disponibile per Micro Focus e per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: somma</p>
BatchJobFailedCount	<p>Il numero di processi non riusciti durante l'intervallo di tempo.</p> <p>Questa metrica è disponibile per Micro Focus e per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: somma</p>
JvmMemoryFree	<p>La quantità di memoria disponibile che non è attualmente utilizzata dalla Java Virtual Machine.</p>

Parametro	Descrizione
	<p>Questa metrica è disponibile solo per il motore di runtime AWS Blu Age. È disponibile per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: byte</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>
JvmMemoryMax	<p>La quantità massima di memoria consentita per la Java Virtual Machine.</p> <p>Questa metrica è disponibile solo per il motore di runtime AWS Blu Age. È disponibile per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: byte</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>
JvmMemoryUsed	<p>La quantità di memoria utilizzata attivamente dalla Java Virtual Machine.</p> <p>Questa metrica è disponibile solo per il motore di runtime AWS Blu Age. È disponibile per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: byte</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>

Parametro	Descrizione
ProcessesActiveCount	<p>Il numero attivo di processi simultanei di esecuzione del servizio che elaborano le richieste.</p> <p>Questa metrica è disponibile solo per il motore di runtime Micro Focus.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: somma</p>
SessionCount	<p>Il numero di sessioni HTTP per l'applicazione.</p> <p>Questa metrica è disponibile solo per il motore di runtime AWS Blu Age. È disponibile per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>
SharedMemoryFree	<p>La memoria disponibile per il server aziendale per archiviare tutte le informazioni necessarie per eseguire transazioni e processi.</p> <p>Questa metrica è disponibile solo per il motore di runtime Micro Focus.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>



Parametro	Descrizione
ThreadActiveCount	<p>Il numero di thread del motore che elaborano le richieste.</p> <p>Questa metrica è disponibile solo per il motore di runtime AWS Blu Age. È disponibile per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>
TransactionCompletedCount	<p>Il numero di transazioni impegnate durante l'intervallo di tempo.</p> <p>Questa metrica è disponibile per Micro Focus e per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: somma</p>
TransactionFailedCount	<p>Il numero di transazioni non riuscite durante l'intervallo di tempo.</p> <p>Questa metrica è disponibile per Micro Focus e per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: numero</p> <p>Statistiche valide: somma</p>

Parametro	Descrizione
TransactionResponseTime	<p>Il periodo di tempo che intercorre tra il momento in cui un utente invia una richiesta e il momento in cui l'applicazione indica che la richiesta è stata completata.</p> <p>Questa metrica è disponibile per Micro Focus e per AWS Blu Age 3.7.0 e versioni successive.</p> <p>Dimensione: ApplicationID</p> <p>Unità: millisecondi</p> <p>Statistiche valide: Average (Media), Minimum, Maximum</p>

## Dimensioni

Dimensione	Descrizione
applicationId	Questa dimensione filtra la metrica in base all'applicazione identificata per ID.
EnvironmentID	Questa dimensione filtra la metrica in base all'ambiente identificato per ID.

## Registrazione delle chiamate all'AWSAPI di modernizzazione del mainframe utilizzando AWS CloudTrail

AWS La modernizzazione del mainframe è integrata con AWS CloudTrail un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o servizio in Mainframe Modernization. AWS AWS CloudTrail acquisisce tutte le chiamate API per AWS la modernizzazione del mainframe come eventi. Le chiamate acquisite includono chiamate dalla console di modernizzazione del AWS mainframe e chiamate in codice alle operazioni dell'API di modernizzazione del AWS mainframe. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli

eventi per AWS la modernizzazione del mainframe. 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, è possibile determinare la richiesta effettuata a AWS Mainframe Modernization, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la Guida per l'[AWS CloudTrailutente](#).

## AWSInformazioni sulla modernizzazione del mainframe in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in AWS Mainframe Modernization, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, vedere [Visualizzazione degli eventi con CloudTrail la cronologia degli eventi](#).

Per una registrazione continua degli eventi del tuo AWS account, inclusi gli eventi per la modernizzazione del AWS mainframe, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni AWS. Il trail registra gli eventi di tutte le Regioni nella partizione AWS e 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](#)
- [Ricezione di file di CloudTrail registro da più account](#)

Tutte le azioni di modernizzazione del AWS mainframe vengono registrate CloudTrail e documentate nel [AWSMainframe](#) Modernization API Reference. Ad esempio, le chiamate a `CreateEnvironment` e le `CreateDeployment` azioni generano `CreateApplication` voci nei file di registro. CloudTrail

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 le credenziali utente root o utente.

- 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](#).

## Informazioni sulle voci dei file di registro di AWS Mainframe Modernization

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'CreateApplicationazione.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAI6WZTHGYAEXAMPLE",
    "arn": "arn:aws:sts::444455556666:assumed-role/Admin/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAI6WZTHGYAEXAMPLE",
        "arn": "arn:aws:iam::444455556666:role/Admin",
        "accountId": "444455556666",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-06-01T20:38:22Z",
        "mfaAuthenticated": "false"
      }
    }
  },
},
```

```
"eventTime": "2022-06-01T20:40:39Z",
"eventSource": "m2.amazonaws.com",
"eventName": "CreateApplication",
"awsRegion": "us-east-1",
"sourceIPAddress": "72.21.196.65",
"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101
Firefox/91.0",
"requestParameters": {
  "clientToken": "1abc23de-f45g-6789-h01i-jkl12m3456789",
  "name": "MyApp",
  "description": "",
  "engineType": "microfocus",
  "definition": {
    "content": "{}"
  },
  "tags": {}
},
"responseElements": {
  "applicationVersion": 1,
  "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
  "applicationArn": "arn:aws:m2:us-east-1:444455556666:app/
lsfhw7fffrosff2lncwqcu",
  "applicationId": "lsfhw7fffrosff2lncwqcu"
},
"requestID": "36982d38-fcde-4bfe-a89a-7bd78d43c926",
"eventID": "d7f0fc36-46ae-4157-9a79-c79f385fda98",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "444455556666",
"eventCategory": "Management"
}
```

# Risoluzione dei problemi

Utilizzate le informazioni contenute in questa sezione per aiutarvi a risolvere gli errori più comuni nelle applicazioni di modernizzazione del AWS mainframe e negli ambienti di runtime utilizzando i motori Blu Age e Micro Focus. AWS

## Argomenti

- [Errore: timeout in attesa che il nome del set di dati venga sbloccato](#)
- [Impossibile accedere all'URL di un'applicazione](#)
- [AWSBlu Insights non si apre dalla console](#)

## Errore: timeout in attesa che il nome del set di dati venga sbloccato

- Motore: AWS Blu Age
- Componente: Blusam

Se vedi questo errore nei CloudWatch log di Amazon per un'applicazione di modernizzazione del AWS mainframe che utilizza il motore AWS Blu Age e funziona in un ambiente con il modello High Availability, indica che un'altra applicazione è bloccata su un set di dati condiviso. In genere, questa situazione si verifica se l'altra applicazione si blocca o fallisce in altro modo e non rilascia il blocco.

## Come si verifica questo errore

L'applicazione `example-app-1` tenta di bloccare un record `example-record-1` per un'operazione di scrittura. Questa operazione crea sia un blocco sul set di dati `example-dataset-1`, che possiede `example-record-1`, sia un blocco su `example-record-1` se stesso. Ora un'altra applicazione prova a bloccare lo stesso record `example-record-1`. `example-app-2` Il set di dati e il record sono già bloccati, quindi `example-app-2` attende il rilascio del blocco. In caso di `example-app-1` arresto anomalo, il blocco mantenuto sul set di dati esiste `example-dataset-1` ancora, il che comporta l'annullamento del tentativo di scrittura e `example-app-2` l'insorgenza di un'eccezione di timeout. Questa situazione di stallo impedisce a tutte le applicazioni di arrivare. `example-dataset-1`

## Come fai a sapere se questa è la tua situazione?

Cerca un'applicazione fallita e controlla se utilizza lo stesso set di dati indicato nel messaggio di errore. Controlla se l'applicazione è in esecuzione in un ambiente di runtime con il modello High Availability. L'applicazione che ha generato l'eccezione di timeout non può procedere e ne visualizzerà lo Failed stato.

## Cosa puoi fare?

Per risolvere immediatamente la situazione, puoi forzare lo sblocco. Per evitare che si verifichi una situazione simile in futuro, puoi configurare due parametri che controllano il meccanismo di riparazione automatica Blusam.

## Forza lo sblocco del lucchetto

Il gestore di blocchi Blusam utilizza Amazon ElastiCache for Redis per fornire blocchi condivisi tra le applicazioni. Per rilasciare i blocchi ElastiCache, utilizzate l'utilità Redis CLI. Non è possibile eliminare un singolo record lock. È necessario rimuovere tutti i blocchi dal set di dati proprietario. Completa questa procedura:


1. Connettiti al tuo ElastiCache usando il seguente comando:

```
redis-cli -h hostname -p port
```

Puoi trovare i dettagli del tuo dispositivo ElastiCache nella ElastiCache console all'[indirizzo https://console.aws.amazon.com/elasticache/](https://console.aws.amazon.com/elasticache/).

2. Inserisci la password.
3. Inserisci il comando che desideri eseguire, come segue:

Comando	Scopo
KEYS *	Ottieni tutte le chiavi esistenti.
CHIAVI * <i>YOUR_DATASET_NAME</i>	Ottieni una chiave di blocco del set di dati.
<i>DEL THE_RETURNED_KEY</i>	Elimina un blocco del set di dati.
FLUSHDB	Pulisci l'intero Redis.

Comando	Scopo
	<div style="border: 1px solid #f08080; padding: 10px; background-color: #fff9f9;"> <p> <b>Warning</b></p> <p>Tutti i dati nella cache Redis andranno persi. Se Redis viene utilizzato per altri scopi, come la gestione delle sessioni http, potresti non volerlo utilizzare. FLUSHDB</p> </div>

## Configura il meccanismo di riparazione automatica Blusam

Il gestore dei blocchi Blusam include un meccanismo di riparazione automatica per prevenire i deadlock su set di dati o record. È possibile regolare i seguenti parametri nella definizione dell'applicazione (`application-main.yml`) per configurare il meccanismo di riparazione automatica:

- `locksDeadTime`: si riferisce al periodo massimo di tempo in cui un'applicazione può mantenere un blocco. Trascorso questo tempo, il blocco viene dichiarato scaduto e rilasciato immediatamente. Il `locksDeadTime` valore è in millisecondi e il valore predefinito è 1000.
- `locksCheck`: definisce la strategia di Blusam locks manager per il controllo dei blocchi. Tutti i lucchetti Blusam hanno una marcatura temporale e una `ElastiCache` scadenza. Il valore del `locksCheck` parametro determina se i blocchi scaduti vengono rimossi.
- `off`: nessun controllo viene eseguito in qualsiasi momento. Potrebbero verificarsi dei deadlock. (Non consigliato)
- `reboot`: i controlli vengono eseguiti quando un'istanza dell'applicazione AWS Mainframe Modernization in esecuzione in un ambiente di runtime AWS Mainframe Modernization viene avviata o riavviata. Tutti i blocchi scaduti vengono rilasciati immediatamente. (Impostazione predefinita)
- `timeout`: i controlli vengono eseguiti quando un'istanza dell'applicazione AWS Mainframe Modernization in esecuzione in un ambiente di runtime AWS Mainframe Modernization viene avviata o riavviata o quando scade un timeout durante un tentativo di bloccare un set di dati. I blocchi scaduti vengono rilasciati immediatamente.



Per ulteriori informazioni sulla definizione dell'applicazione per un'applicazione AWS Blu Age, vedere [AWS Esempio di definizione dell'applicazione Blu Age](#)

## Blusam locks manager

Nel contesto di un ambiente di runtime di modernizzazione del AWS mainframe che utilizza il modello High Availability, un'applicazione AWS Blu Age potrebbe essere implementata più volte. Per le applicazioni che gestiscono set di dati Blusam, potrebbero verificarsi problemi di accesso simultaneo. Il gestore di blocchi Blusam garantisce l'integrità dei dati e gestisce l'accesso in lettura e scrittura a record e set di dati fornendo blocchi condivisi tra le applicazioni che utilizzano. ElastiCache Questo meccanismo consente a più di un'applicazione di leggere il record contemporaneamente e garantisce che solo un'applicazione alla volta scriva il record.

### Scrivi lucchetti

Per aggiornare o eliminare un record specifico, l'applicazione deve prima bloccare il set di dati proprietario del record, quindi bloccare il record stesso. Quando il record è bloccato, il blocco del set di dati viene rilasciato e altri record dello stesso set di dati sono disponibili per l'uso. Al termine dell'operazione di aggiornamento o eliminazione, viene rilasciato il blocco dei record mantenuto. Solo un'applicazione alla volta può aggiornare il record, il che impedisce alle altre applicazioni di leggere o scrivere fino al rilascio del blocco, se la politica dell'applicazione definita consente l'attesa del rilascio.

### Leggi i blocchi

Finché non viene mantenuto alcun blocco di scrittura sul record o sul set di dati, più applicazioni possono leggere gli stessi record contemporaneamente. Per bloccare un record per un'operazione di scrittura, tutti i blocchi di lettura devono essere rilasciati.

#### Note

Il gestore dei blocchi Blusam gestisce l'accesso da più thread in una determinata applicazione utilizzando lo stesso meccanismo di blocco.

## Impossibile accedere all'URL di un'applicazione

- Motore: AWS Blu Age e Micro Focus
- Componente: applicazioni

Se non riesci ad accedere all'URL di un'applicazione di modernizzazione del AWS mainframe in esecuzione che hai creato e distribuito in un ambiente di runtime di modernizzazione del AWS mainframe, potresti dover configurare le regole in entrata sul gruppo di sicurezza associato all'ambiente di runtime.

## Come si verifica questo errore

Quando si crea un ambiente di runtime, il gruppo di sicurezza fornito, incluso il gruppo di sicurezza predefinito, deve disporre di regole in entrata configurate per consentire il traffico verso le applicazioni distribuite dall'esterno del VPC, se si desidera consentire questo tipo di accesso.

## Come fai a sapere se questa è la tua situazione?

L'applicazione è stata avviata correttamente e funziona normalmente, ma non è possibile connettersi utilizzando l'URL.

## Cosa puoi fare?

Verifica se il gruppo di sicurezza Amazon VPC associato all'ambiente di runtime consente il traffico verso l'ambiente sulle porte applicative appropriate. Per verificare le regole del gruppo di sicurezza, completa i seguenti passaggi:

1. [Apri la console di modernizzazione AWS del mainframe all'indirizzo https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Nella barra di navigazione a sinistra, scegli Ambienti.
3. Scegli l'ambiente di runtime che ospita l'applicazione a cui desideri connetterti.
4. Scegli Configurazioni.
5. In Sicurezza e rete, scegli il gruppo di sicurezza. Il link apre i dettagli del gruppo di sicurezza nella console Amazon VPC.
6. Se necessario, scegli Modifica regole in entrata e aggiungi la seguente regola se non è già presente:

Type

TCP personalizzato

## Porta

8196 o la porta che corrisponde alle proprietà del listener specificate nella definizione dell'applicazione. Per ulteriori informazioni, consulta [Fase 2: Creare la definizione dell'applicazione](#).

## Origine

L'indirizzo IP da cui si chiama l'applicazione. Puoi scegliere MyIP dal menu a discesa. Se hai ancora problemi di timeout, prova a scegliere Anywhere IPV4 o Anywhere IPV6. Assicurati di arrestare l'applicazione e riavviarla dopo aver aggiunto la regola in entrata al gruppo di sicurezza.

Per ulteriori informazioni, consulta [Lavora con le regole dei gruppi di sicurezza](#) nella Guida per l'utente di Amazon VPC.

## AWSBlu Insights non si apre dalla console

- Motore: AWS Blu Age
- Componente: Blu Insights

Quando si tenta di accedere a Blu Insights dalla console di modernizzazione del AWS mainframe, questa non si apre e la nuova scheda viene chiusa immediatamente.

### Come si verifica questo errore

Il ruolo che stai utilizzando per accedere a Blu Insights non dispone di autorizzazioni sufficienti.

### Cosa puoi fare?

Allega una policy IAM al ruolo per consentirgli di accedere a Blu Insights. Assicurati che la policy includa almeno le seguenti autorizzazioni.

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

```
    "Action": [  
      "m2:GetSignedBluinsightsUrl"  
    ],  
    "Resource": "arn:aws:m2:region:account:*"  
  }  
]  
}
```

Assicurati di sostituire `region` e `account` con la corretta Regione AWS e Account AWS.

# Cronologia dei documenti per la guida per l'AWSutente alla modernizzazione del mainframe

La tabella seguente descrive le versioni della documentazione per la modernizzazione del AWS mainframe.

Modifica	Descrizione	Data
<a href="#">Tutorial aggiornato su Managed Runtime for Micro Focus</a>	Questo tutorial mostra come distribuire ed eseguire l'applicazione di CardDemo esempio in un ambiente di runtime gestito dalla modernizzazione del AWS mainframe con il motore di runtime Micro Focus.	5 febbraio 2024
<a href="#">Note di rilascio per AWS Blue Age Runtime and Modernization Tools versione 3.9.0.</a>	Questa versione di AWS Blue Age Runtime and Modernization Tools si concentra su diversi miglioramenti trasversali del prodotto, volti ad aumentare le prestazioni nelle architetture ad alta disponibilità, insieme a nuove funzionalità per portare l'esecuzione dei lavori a un livello superiore.	18 dicembre 2023
<a href="#">Trasferisci file tra mainframe e AWS</a>	È stata rilasciata una nuova funzionalità per il trasferimento di file dal mainframe di origine a. AWS	27 novembre 2023
<a href="#">Gestisci le transazioni per le applicazioni</a>	È stata rilasciata una nuova funzionalità per visualizzare e modificare le transazioni per le	16 ottobre 2023

---

<a href="#"><u>Note di rilascio per AWS Blu Age Runtime and Modernization Tools versione 3.6.0.</u></a>	applicazioni per la modernizzazione del AWS mainframe.  Questa versione di AWS Blu Age Runtime and Modernization Tools offre nuove funzionalità per le migrazioni precedenti di zOS e AS400, principalmente orientate all'espansione dei meccanismi di supporto CICS, all'integrazione delle funzionalità JCL, all'ottimizzazione delle prestazioni nelle funzionalità simultanee e ad alto volume e all'aggiunta di funzionalità multi-data-source	4 agosto 2023
<a href="#"><u>È ora possibile distribuire una nuova versione di un'applicazione quando l'applicazione viene interrotta.</u></a>	In precedenza, per distribuire una nuova versione di un'applicazione, era necessario eliminare la versione distribuita. Ora puoi semplicemente interrompere la versione distribuita e distribuire una nuova versione.	26 luglio 2023
<a href="#"><u>AWSRuntime Blu Age confezionato per una distribuzione più semplice di Amazon EC2</u></a>	AWSLa modernizzazione del mainframe con il runtime AWS Blu Age è ora disponibile con maggiore flessibilità per la configurazione dello stack completo e la distribuzione sulle istanze Amazon EC2 del tuo Account AWS	6 luglio 2023

---

<a href="#">Accesso singolo a Blu Age Blu Insights. AWSAWS</a>	AWSAWSBlu Age Blu Insights è disponibile AWS Management Console tramite Single Sign-On.	31 marzo 2023
<a href="#">Versione GA</a>	Versione GA della Guida per l'utente di AWS Mainframe Modernization.	8 giugno 2022
<a href="#">Versione iniziale</a>	Versione iniziale (anteprima pubblica) della AWS Mainframe Modernization User Guide.	30 novembre 2021

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.