Guida per l'utente

AWS Modernizzazione del mainframe



Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Modernizzazione del mainframe: Guida per l’utente

Copyright © 2026 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 dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

..... xv
Che cos'è la modernizzazione del AWS mainframe 1
Caratteristiche AWS della modernizzazione del mainframe 2
Pattern 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 per la modernizzazione del mainframe AWS 5
AWS Modernizzazione del mainframe: modifica della disponibilità 6
Configurazione per la modernizzazione del AWS mainframe 8
Registrati per un Account AWS 8
Crea un utente con accesso amministrativo 8
Concetti 11
Applicazione 11
Definizione dell'applicazione 11
Lavoro in batch 12
Configurazione 13
Set di dati 13
Ambiente 13
Modernizzazione del mainframe 13
Viaggio di migrazione 13
Punto di montaggio 13
Rifattorizzazione automatizzata 14
Ridefinizione della piattaforma 14
Risorsa 14
Motore di runtime 14
Approccio alla modernizzazione 15
Fase di valutazione 15
Fase di mobilitazione 15
Fase di migrazione e modernizzazione 16
Fase operativa e di ottimizzazione 16
Nozioni di base 17
Tutorial: configurare il runtime gestito per AWS Transform for mainframe 17

Prerequisiti 18
Passaggio 1: carica l'applicazione demo 18
Fase 2: Creare la definizione dell'applicazione 18
Fase 3: Creare un ambiente di runtime 19
Fase 4: creare un'applicazione 24
Fase 5: Implementazione di un'applicazione 26
Fase 6: Avviare un'applicazione 28
Passaggio 7: Accedere all'applicazione 28
Fase 8: test dell'applicazione 29
Eseguire la pulizia delle risorse 31
Tutorial: configurare il runtime gestito per Rocket Software 31
Prerequisiti 32
Fase 1: creare e caricare un bucket Amazon S3 32
Passaggio 2: creare e configurare un database 34
Passaggio 3: creare e configurare un AWS KMS key 36
Passaggio 4: Creare e configurare un Gestione dei segreti AWS database segreto 37
Passaggio 5: aggiungere SSLMode al segreto 38
Fase 6: Creare un ambiente di runtime 39
Fase 7: Creare un'applicazione 44
Fase 8: Implementazione di un'applicazione 50
Fase 9: Importazione dei set di dati 52
Fase 10: Avviare un'applicazione 58
Fase 11: Connect all'applicazione CardDemo CICS 59
Eseguire la pulizia delle risorse 66
Fasi successive 67
Ciclo di vita dei componenti 68
Panoramica del ciclo di vita dei componenti 68
Aggiornamento della versione 69
AWS Panoramica della versione di Mainframe Modernization Refactor with Transform for mainframe AWS 71
Applicazioni gestite 72
Crea risorse per un'applicazione migrata AWS 73
Autorizzazioni richieste 73
Bucket Amazon S3 73
Database 74
AWS Key Management Service chiave 75

Gestione dei segreti AWS segreto 75
Creazione di un'applicazione 76
Creazione di un'applicazione 76
Implementa un'applicazione 77
Distribuisci un'applicazione 77
Aggiornamento di un'applicazione 78
Aggiornamento di un'applicazione 79
Eliminazione di un'applicazione 79
Eliminazione di un'applicazione 80
Invia lavori in batch per le candidature 80
Invia un lavoro in batch 81
Riavviare un processo batch 82
Annulla i processi in batch per le applicazioni 83
Annullamento di un processo in batch 84
Importa set di dati per applicazioni 84
Importa un set di dati 85
Esporta set di dati per le applicazioni 85
Esportazione di un set di dati 86
Gestisci le transazioni per le applicazioni 86
Gestisci le transazioni per le applicazioni 87
Configurare l'applicazione gestita da Rocket Software 88
Integrazioni di terze parti supportate per Rocket Software 88
Configurare l' AWS applicazione gestita Transform for mainframe 91
Struttura di AWS Transform per applicazioni gestite da mainframe 91
Configurare l'accesso alle utilità per le applicazioni gestite 93
Configura proprietà aggiuntive per l'applicazione gestita 104
Riferimento alla definizione dell'applicazione 125
Sezione generale dell'intestazione 126
Panoramica della sezione relativa alle definizioni 128
AWS Esempio di definizione di applicazione Transform for mainframe 128
AWS Transform for mainframe: dettagli della definizione 129
Definizione dell'applicazione Rocket Software 135
Dettagli della definizione di Rocket Software 137
Data set definition reference 147
Proprietà comuni 148
Esempio di formato di richiesta di set di dati per VSAM 150

Esempio di formato di richiesta del set di dati per GDG base 152
Esempio di formato di richiesta del set di dati per le generazioni PS o GDG 153
Esempio di formato di richiesta del set di dati per PO 154
Ambienti di runtime gestiti 157
Creare un ambiente di runtime 158
Crea un ambiente di runtime 158
Aggiornare un ambiente di runtime 161
Aggiorna un ambiente di runtime 162
Finestra di manutenzione 162
Arresta un ambiente di runtime 163
Arresta un ambiente di runtime 163
Riavviare un ambiente di runtime 164
Riavvia un ambiente di runtime 164
Eliminare un ambiente di runtime 165
Eliminare un ambiente di runtime 165
AWS Transform per il refactoring del mainframe 167
AWS Transform per versioni mainframe 168
AWS Transform per il controllo delle versioni mainframe 168
AWS Transform for mainframe Runtime 169
AWS Note sulla versione di Transform for mainframe 171
AWS Trasforma per le vulnerabilità di sicurezza del mainframe 316
Aggiornamento di Transform per mainframe AWS 318
AWS Trasformazione per il ciclo di vita del mainframe 321
AWS Concetti di Transform for mainframe Runtime 322
Architettura di alto livello 323
Struttura applicativa modernizzata 328
Comprendi i semplificatori di dati 364
AWS Transform per mainframe Blusam 372
BlusamConsole di amministrazione 396
AWS Configurazione Transform for mainframe Runtime 436
Nozioni di base sulla configurazione delle applicazioni 437
Precedenza delle applicazioni 439
JNDI per database 439
AWS I segreti di Transform for mainframe Runtime 440
Altri file (groovy, sql, ecc.) 453
Applicazione web aggiuntiva 453

Abilita le proprietà 454
Proprietà della cache Redis disponibili 568
Configurare la sicurezza per le applicazioni Gapwalk 586
AWS Codici di errore di runtime Transform for mainframe 607
Codici di errore generici (BA-A) 608
Codici di errore relativi a Blusam (BA-B) 614
Codici di errore relativi ai batch (BA-C) 615
Codici di errore relativi a Datasimplifier (BA-D) 615
Codici di errore relativi ai file (BA-F) 616
Codici di errore relativi a CL Command Programs (BA-H) 616
Codici di errore relativi a JICS (BA-J) 617
Codici di errore relativi a ADABAS (BA-N) 618
Codici di errore relativi a Queue (BA-Q) 618
Codici di errore relativi a Redis (BA-R) 619
Codici di errore relativi ai programmi di utilità (BA-U) 649
AWS Transform for mainframe Runtime APIs 657
Endpoint per la costruzione URLs 658
Endpoint per l'applicazione Gapwalk 658
Blusamendpoint REST della console delle applicazioni 683
Gestisci la console dell'applicazione JICS 706
Strutture di dati 728
AWS Transform per mainframe Runtime Utilities 737
Configura e distribuisci l'applicazione web Utility 737
Programmi di utilità disponibili 737
Utilità per set di dati 739
Utilità del database 756
Comandi e utilità 767
Utilità di ordinamento 771
Altro/Utilità varie 783
Configurare AWS Transform for mainframe Runtime 795
AWS Prerequisiti di Transform for mainframe Runtime 796
Onboarding AWS Transform per mainframe Runtime 797
Requisiti di configurazione dell'infrastruttura 802
AWS Transform for mainframe Runtime artefact 805
Implementa AWS Transform for mainframe Runtime su Amazon EC2 807
Implementa AWS Transform for mainframe Runtime su Amazon ECS e Amazon EKS 818

Prova l' PlanetsDemo applicazione 826
Modifica il codice sorgente con AWS Transform for mainframe Developer IDE 830
Tutorial: Configurazione WorkSpaces delle applicazioni per AWS Transform for mainframe Developer IDE 830
Tutorial: Usa AWS Transform for mainframe Developer sulle applicazioni WorkSpaces 836
AWS Domande frequenti su Transform for mainframe 853
Ambito generale 853
AWS Transform for mainframe Runtime 855
Dati 865
Trasformazione 866
Implementazione 867
Sicurezza 869
Rocket Software Replatforming 872
Configurazione del software Rocket (su Amazon EC2) 872
Prerequisiti del software Rocket (su Amazon EC2) 873
Crea l'endpoint Amazon VPC per Amazon S3 873
Richiedi l'aggiornamento della lista consentita per l'account 876
Crea il AWS Identity and Access Management ruolo 877
Concedi a License Manager le autorizzazioni richieste 884
Iscriviti ad Amazon Machine Images 885
Avviate un'istanza di Rocket Software 889
Subnet o VPC senza accesso a Internet 896
Configurare l' WorkSpaces automazione delle applicazioni 902
Configurare l'automazione all'inizio della sessione 903
Configura l'automazione al termine della sessione 903
Visualizza i set di dati come tabelle in Enterprise Developer 903
Prerequisiti 904
Passaggio 1: configurare la connessione ODBC al datastore di Rocket Software (database Amazon RDS) 904
Fase 2: Creare il file MFDBFH.cfg 907
Fase 3: Create un file di struttura (STR) per il layout del quaderno 907
Fase 4: Creare una visualizzazione del database utilizzando il file di struttura (STR) 910
Fase 5: Visualizzate i set di dati di Rocket Software (precedentemente Micro Focus) come tabelle e colonne 910
Modifica i set di dati utilizzando Data File Tools in Enterprise Developer 911
Prerequisiti 912

Avvia Rocket Software (precedentemente Micro Focus) Data File Tools 912
Modifica i set di dati VSAM memorizzati nel database MFDBFH 913
Modifica i set di dati non VSAM memorizzati nel database MFDBFH 916
Modifica i set di dati VSAM e non VSAM memorizzati nel file system (EFS/FSx) 918
Tutorial per Rocket Software 918
Tutorial: configura la build per l'applicazione di esempio BankDemo 919
Tutorial: Configurazione WorkSpaces delle applicazioni per Enterprise Analyzer ed Enterprise Developer 928
Tutorial: Usa i modelli con Rocket Enterprise Developer 937
Tutorial: configurare Enterprise Analyzer 948
Tutorial: configurare Enterprise Developer 959
Utilità Batch 965
Ubicazione binaria 966
Utilità batch M2SFTP 966
Utilità batch M2WAIT 973
TXT2Utilità PDF batch 975
Utilità batch M2DFUTIL 981
Utilità batch M2RUNCMD 988
Trasferimento di file 992
Che cos'è il trasferimento di file 992
Vantaggi del trasferimento di file per la modernizzazione dei mainframe di AWS 993
Come funziona AWS Mainframe Modernization File Transfer 993
Installa un agente di trasferimento file 994
Fase 1: Creare un set di dati ZfS per M2-Agent 995
Fase 2: Formattare il set di dati come ZFs 995
Fase 3: Montare il filesystem 995
Fase 4: Verificare il montaggio 996
Fase 5: Inserire OMVS 996
Passaggio 6: Impostare la variabile di ambiente della directory di installazione dell'agente .. 996
Fase 7: Impostare la variabile di ambiente della directory di lavoro 996
Fase 8: Creare la cartella di lavoro 997
Fase 9: Copiare il file tar dell'agente e copiare la directory di lavoro 997
Fase 10: Completare l'installazione dell'agente 997
Configurare un agente di trasferimento file 998
Passaggio 1: Configurazione delle autorizzazioni e Started Task Control (STC) 998
Fase 2: creazione di bucket Amazon S3 999

Fase 3: Creare una chiave di crittografia gestita AWS KMS dal cliente 1000
Fase 4: Creare un Gestione dei segreti AWS segreto per le credenziali del mainframe 1001
Fase 5: Creare una policy IAM 1002
Fase 6: Creare un utente IAM con credenziali di accesso a lungo termine 1003
Fase 7: Creare un ruolo IAM che l'agente deve assumere 1004
Fase 8: Configurazione dell'agente 1005
Crea endpoint per il trasferimento dei dati 1008
Crea endpoint per il trasferimento dei dati 1008
Crea attività di trasferimento 1010
Crea attività di trasferimento 1010
Visualizza le attività di trasferimento 1014
Tutorial: Guida introduttiva a File Transfer 1015
Panoramica di 1015
Passaggio 1: trasferire il pacchetto tar dei binari dell'agente dalla partizione logica del mainframe AWS 1015
Fase 2: Configurare l'agente File Transfer sul mainframe di origine 1015
Fase 3: Creare un endpoint per il trasferimento dei dati 1016
Passaggio 4: Creare un'attività di trasferimento 1016
Passaggio 5: Visualizza lo stato di avanzamento dell'operazione di trasferimento 1016
Pagine di codice sorgente e di destinazione supportate 1016
Tipi di set di dati mainframe 1016
Pagine codici supportate 1017
AWS Transform per mainframe 1018
Vantaggi principali 1018
Procedura dettagliata sulla trasformazione della console delle applicazioni mainframe 1019
Replica dei dati con Precisly 1020
Prerequisiti 1020
Iscriviti all'Amazon Machine Image 1020
Avvia la replica dei dati di AWS Mainframe Modernization con Precisly 1021
Creazione di una policy IAM 1022
Creazione di un ruolo IAM 1023
Collega il ruolo IAM all'istanza Amazon EC2 1023
Conversione di assembler con MLogica 1024
Cos'è la conversione di Assembler con MLogica 1024
Compilatori di conversione del codice 1025
Architettura di conversione del codice 1025

Approccio all'automazione 1026
Sicurezza 1026
Risorse aggiuntive 1027
Comprendi la fatturazione con conversione del codice 1027
Conversione, fatturazione e ambito del codice. 1027
Concetti di conversione del codice 1029
Gestione delle macro 1030
Pagine codici (EBCDIC o ASCII) 1030
CodeBuild 1030
Comprendi i componenti e i processi 1030
Modernizzazione del mainframe AWS contenitore 1031
Bucket per progetti S3 1031
Posizioni dei file di registro 1032
Panoramica del processo 1032
Tutorial: converti il codice da Assembler a COBOL 1033
Prerequisiti 1034
Passaggio 1: condividi le risorse di compilazione con Account AWS 1034
Fase 2: creazione di bucket Amazon S3 1034
Fase 3: Creare una policy IAM 1035
Fase 4: Creare un ruolo IAM 1037
Fase 5: collegare la policy IAM al ruolo IAM 1038
Fase 6: Creare il progetto CodeBuild 1038
Fase 7: Definizione del progetto e caricamento del codice sorgente 1044
Fase 8: Esegui l'analisi e comprendi i report 1046
Passaggio 9: Esegui la conversione del codice 1048
Passaggio 10: verifica la conversione del codice 1051
Passaggio 11: scarica il codice convertito 1052
Eseguire la pulizia delle risorse 1053
Integrazione con Charon 1054
Introduzione a Charon-SSP 1054
Sistemi operativi guest supportati 1056
Prerequisiti dell'istanza cloud Charon-SSP 1056
Prerequisiti dell'istanza 1058
Creazione e configurazione di un'istanza AWS cloud per Charon (Nuova GUI) 1059
Prerequisiti generali 1059
Utilizzo di Console di gestione AWS per avviare una nuova istanza 1061

Ripiattaforma con NTT DATA 1066
Prerequisiti 1066
Iscriviti all'Amazon Machine Image 1066
Avvia la ripiattaforma AWS Mainframe Modernization con l'istanza NTT DATA 1067
Guida introduttiva a NTT Data 1067
Tutorial: distribuisci l'applicazione su NTT DATA CardDemo 1069
Diagramma di flusso di distribuzione 1069
Prerequisiti 1070
Fase 1: Preparare l'ambiente 1071
Fase 2: Creare una regione TPE 1071
Fase 3: Creare il nodo e il sottosistema BPE 1072
Fase 4: Compilare e distribuire l'applicazione CardDemo 1081
Fase 5: Importazione del catalogo BPE e TPE 1083
Fase 6: Avviare e collegare TPE con BPE 1083
Fase 7: Eseguire l'applicazione CardDemo 1084
Risoluzione dei problemi 1090
Sicurezza 1092
Protezione dei dati 1093
Dati raccolti da AWS Mainframe Modernization 1094
Crittografia dei dati inattiva per il servizio di modernizzazione AWS del mainframe 1095
In che modo AWS Mainframe Modernization utilizza le sovvenzioni in AWS KMS 1097
Creazione di una chiave gestita dal cliente 1099
Specificazione di una chiave gestita dal cliente per la modernizzazione del mainframe AWS 1101
AWS Modernizzazione del mainframe: contesto di crittografia 1102
Monitoraggio delle chiavi di crittografia 1103
Ulteriori informazioni 1119
Crittografia dei dati in transito 1119
Identity and Access Management 1120
Destinatari 1120
Autenticazione con identità 1121
Gestione dell'accesso tramite policy 1122
Come funziona la modernizzazione AWS del mainframe con IAM 1124
Esempi di policy basate su identità 1138
Risoluzione dei problemi 1141
Uso di ruoli collegati ai servizi 1143

Convalida della conformità 1146
Resilienza 1147
Sicurezza dell'infrastruttura 1148
AWS PrivateLink 1148
Considerazioni 1149
Creazione di un endpoint di interfaccia 1149
Creazione di una policy dell'endpoint 1149
Monitoraggio 1151
Monitoraggio con CloudWatch 1151
Metriche dell'ambiente di runtime 1152
Parametri di applicazioni 1153
Dimensioni 1159
Registrazione delle chiamate API di CloudTrail con 1159
AWS Informazioni sulla modernizzazione del mainframe in CloudTrail 1160
Informazioni sulle voci dei file di registro di AWS Mainframe Modernization 1161
Risoluzione dei problemi in M2 1163
Errore di risoluzione dei problemi: timeout in attesa dello sblocco del nome del set di dati 1163
Cause comuni 1164
Risoluzione 1164
Forza il rilascio della serratura 1164
Configura il Blusam meccanismo di riparazione automatica 1165
Blusamgestore delle serrature 1166
Errore di risoluzione dei problemi: impossibile accedere all'URL di un'applicazione 1167
Cause comuni 1167
Risoluzione 1167
Risoluzione dei problemi: AWS Transform for mainframe refactor non si apre dalla console ... 1168
Cause comuni 1168
Risoluzione 1169
Errore di risoluzione dei problemi: ambiente non integro 1169
Cause comuni 1170
Risoluzione 1170
Risoluzione dei problemi di licenza per Rocket Software 1171
Verifica che l'istanza Amazon EC2 abbia il ruolo di licenza IAM 1171
Usa l'analizzatore di raggiungibilità 1172
Esegui il license-daemon 1172

Problemi di licenza con Enterprise Server o Enterprise Build Tools su Linux dopo l'applicazione di patch al sistema operativo 1173
Cronologia dei documenti 1175

AWS Il servizio di modernizzazione del mainframe (esperienza Managed Runtime Environment) non è più aperto a nuovi clienti. Per funzionalità simili a AWS Mainframe Modernization Service (esperienza Managed Runtime Environment), esplora AWS Mainframe Modernization Service (Self-Managed Experience). I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Modifica della disponibilità di Mainframe Modernization.

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.

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 oppure PL/I implementare una pipeline automatizzata per l'integrazione e la distribuzione continue delle applicazioni. (CI/CD) 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
- Pattern
- 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 per la modernizzazione del mainframe AWS

> **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 l'ambito e pianificare un progetto di migrazione e modernizzazione.
- Refactor: grazie alla tecnologia AWS Transform for mainframe, puoi utilizzare il refactoring per convertire i linguaggi di programmazione delle applicazioni esistenti, creare macroservizi o microservizi e modernizzare le interfacce utente () e gli stack di software applicativi. UIs

  AWS Transform for mainframe refactor è ora disponibile tramite Single Sign-On. Console di gestione AWS Non è più necessario gestire credenziali di refactoring separate per AWS Transform for mainframe. È possibile accedere alle funzionalità AWS Transform for mainframe Codebase e Transformation Center direttamente da. Console di gestione AWS
- Replatform: grazie alla soluzione Micro Focus Enterprise, è possibile portare 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 runtime gestito 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 CI/CD funzionalità 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: la modernizzazione del AWS mainframe supporta AWS CloudFormation e consente un'implementazione ripetibile e una AWS PrivateLink maggiore sicurezza e conformità 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, consulta Funzionalità di modernizzazione del mainframe.AWS

Pattern

Il modello Automated Refactoring, basato su AWS Transform for mainframe, 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 data store. 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 Rocket Software (precedentemente Micro Focus) Enterprise, è incentrato 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

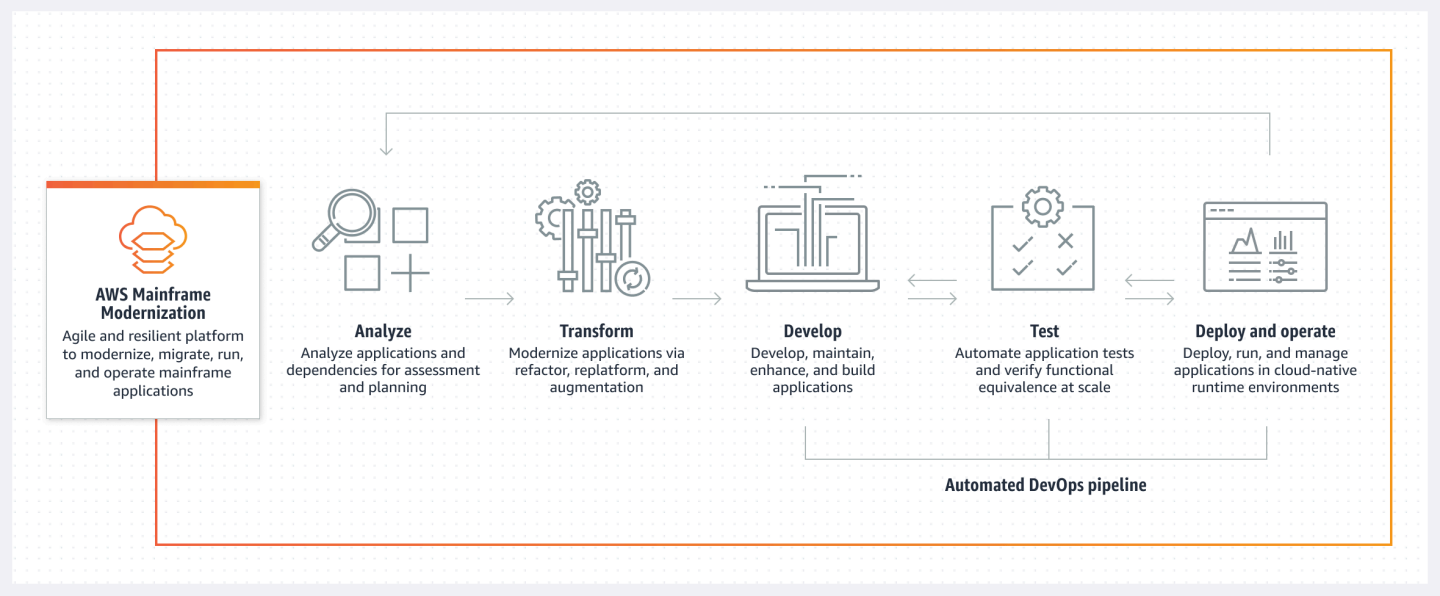
Eseguire una prova Offriamo tutorial ed esempi di applicazioni per aiutarti a farti un'idea di ciò AWS che offre la modernizzazione del mainframe. Scegli il Tutorial: configurare il runtime gestito per AWS Transform for mainframe o il Tutorial: Configurazione del runtime gestito per Rocket Software (precedentemente Micro Focus) per un tutorial completo. step-by-step

Se sei interessato al refactoring automatizzato, dai un'occhiata agli strumenti AWS Transform for mainframe su AWS Transform for mainframe refactor. Puoi anche configurare WorkSpaces le applicazioni per accedere all'IDE AWS Transform for mainframe Developer o agli strumenti Rocket

Enterprise Analyzer (precedentemente Micro Focus Enterprise Analyzer) e Rocket Enterprise Developer (precedentemente Micro Focus Enterprise Developer).

I tutorial e le applicazioni di esempio ti danno solo un'idea di ciò che offre la modernizzazione del mainframe. AWS 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.

Il diagramma seguente mostra il flusso di lavoro del servizio di modernizzazione del AWS mainframe per analizzare, trasformare, sviluppare, testare, implementare e utilizzare le applicazioni mainframe.



Servizi correlati

Oltre a AWS Transform for mainframe refactor per il refactoring automatizzato, è possibile utilizzare i seguenti servizi con Mainframe Modernization. AWS AWS

- Amazon RDS per l'hosting dei 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 Rocket Enterprise Analyzer e Rocket Enterprise Developer
- CloudFormation per la DevOps pipeline automatizzata che puoi utilizzare per configurare le tue applicazioni migrate CI/CD
- 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/ 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:

- Inizia con la modernizzazione del AWS mainframe
- Configurazione per la modernizzazione del AWS mainframe

Prezzi per la modernizzazione del mainframe AWS

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 Transform for mainframe refactor, paghi per l'utilizzo di Transformation Center. Per ulteriori informazioni, consulta i prezzi di AWS Mainframe Modernization.

AWS Modernizzazione del mainframe: modifica della disponibilità

AWS Il Mainframe Modernization Service (Managed Runtime Environment experience) ha smesso di accettare nuovi clienti, mentre i clienti esistenti possono continuare a utilizzare il servizio senza interrompere i progetti di modernizzazione in corso. Questo servizio fornisce versioni gestite e autogestite della soluzione software di terze parti Rocket per il replatforming. Questa modifica della disponibilità del servizio riguarda solo la versione Managed. I nuovi progetti verranno indirizzati alla versione autogestita.

Sebbene non aggiungeremo nuove funzionalità al servizio, rimaniamo impegnati a fornire aggiornamenti di sicurezza e a mantenere la disponibilità del servizio per garantire che i progetti di modernizzazione in corso continuino a funzionare senza intoppi. Il nostro obiettivo è garantire un ambiente stabile in cui i clienti esistenti possano completare le iniziative di modernizzazione del mainframe preparandosi al contempo alla transizione alla versione autogestita.

AWS Il servizio di modernizzazione del mainframe (Self-Managed Experience) è la nostra soluzione consigliata che offre le stesse funzionalità di base per la modernizzazione del mainframe.

Se avete altre domande, contattateci tramite il o leggete il nostro Supporto AWS FAQs

1. Cosa significa questo per il servizio (avete intenzione di chiudere il servizio?)? AWS Mainframe Modernization Service (Managed Runtime Environment experience) ha smesso di accettare nuovi clienti. Il servizio continuerà a funzionare per consentire ai clienti esistenti di completare i loro progetti di migrazione in corso.
2. In che modo verranno influenzati i clienti esistenti? I clienti esistenti continueranno a ricevere supporto completo, con AWS Proserve disponibile ad assisterli se scelgono di passare al runtime autogestito. Contatta Supporto AWSper ricevere assistenza.
3. Se un cliente esistente ha dei problemi, come possono risolverli? I clienti esistenti di servizi gestiti possono continuare a lavorare con AWS Support per risolvere i problemi e i AWS team continueranno a supportare i clienti.
4. Quali sono le alternative che i clienti possono esplorare? La versione autogestita di AWS Mainframe Modernization Service offre funzionalità di runtime sia per Rocket Software (replatform) che per AWS Transform for mainframe (refactor).
5. In che modo i clienti possono migrare dal AWS Mainframe Modernization Service (esperienza Managed Runtime Environment)? I clienti che oggi utilizzano il Managed Runtime continueranno

a ricevere assistenza. Per migrare gli ambienti esistenti, AWS ProServe è possibile richiedere assistenza nella transizione all'utilizzo di opzioni autogestite.

Configurazione per la modernizzazione del AWS mainframe

Prima di iniziare a utilizzare AWS Mainframe Modernization, tu o il tuo amministratore dovete registrarvi Account AWS, creare un utente con impostazioni amministrative e proteggere gli utenti IAM.

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Registrati per un Account AWS

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

Per iscriverti a un Account AWS

1. Apri la https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente 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. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a https://aws.amazon.com/e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato 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 Console di gestione AWScome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

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

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

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

   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.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente di AWS IAM Identity Center .

AWS Concetti di modernizzazione del mainframe

AWS Mainframe Modernization fornisce strumenti e risorse per aiutarti a migrare, modernizzare ed eseguire carichi di lavoro mainframe su. AWSÈ possibile utilizzare questa pagina per comprendere vari concetti relativi alla modernizzazione del AWS mainframe, tra cui applicazioni, modernizzazione, ambienti, replatforming, refactoring e motori di runtime.

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 viene 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 per l'utilizzo da parte delle 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.

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 un secondo momento. 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 Transform for mainframe 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 di Rocket Software (precedentemente 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 siete pronti 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 e 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 ripiattaforma o il refactoring del codice, l'integrazione con altri sistemi e, ovviamente, i test. Al termine della fase, distribuisci 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 automatizzato, consulta AWS Transform for mainframe refactor. AWS Transform for mainframe refactor è ora disponibile tramite Single Sign-On. Console di gestione AWS Non è più necessario gestire credenziali di refactoring separate per AWS Transform for mainframe. È possibile accedere alle funzionalità AWS Transform for mainframe Codebase e Transformation Center direttamente da. Console di gestione AWS

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.

Inizia con la modernizzazione del AWS mainframe

Puoi iniziare a usare AWS Mainframe Modernization seguendo i tutorial che ti introducono al servizio e a ogni motore di runtime.

Argomenti

- Tutorial: configurare il runtime gestito per AWS Transform for mainframe
- Tutorial: Configurazione del runtime gestito per Rocket Software (precedentemente Micro Focus)

Per continuare a imparare, consulta i seguenti tutorial.

- Tutorial: configurazione della build Rocket Software (precedentemente Micro Focus) per l'applicazione di esempio BankDemo
- Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket Enterprise Developer

Tutorial: configurare il runtime gestito per AWS Transform for mainframe

È possibile implementare un'applicazione modernizzata AWS Transform for mainframe in un ambiente di runtime di modernizzazione del AWS mainframe con un'applicazione demo specificata in questo tutorial.

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

- Eseguire la pulizia delle risorse

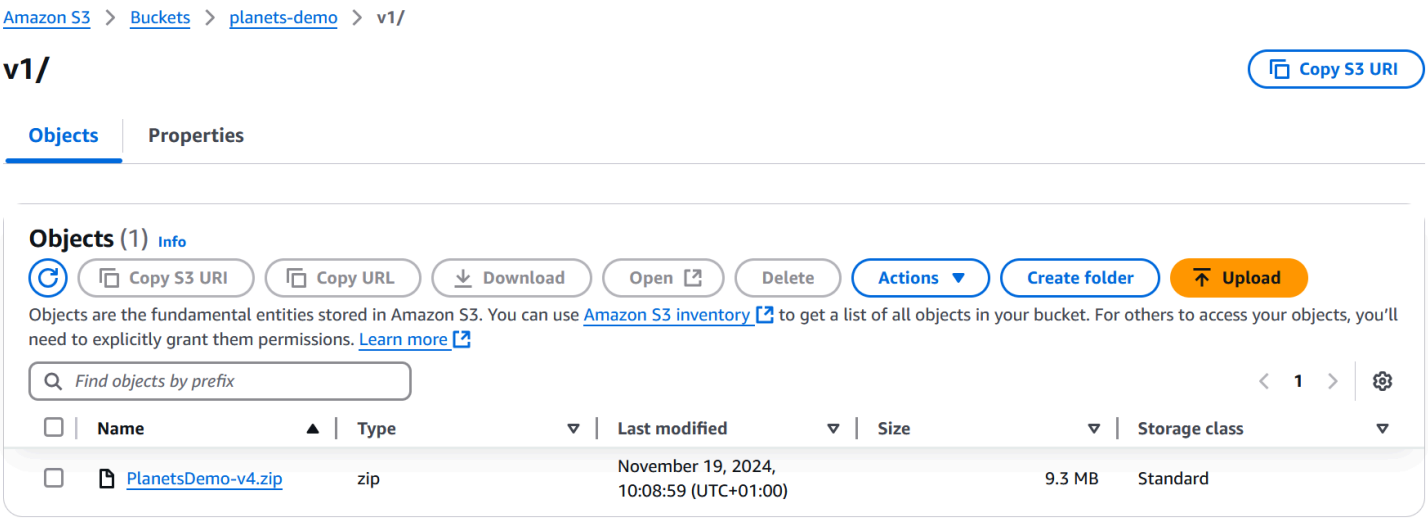
Prerequisiti

Per completare questo tutorial, scaricate l'archivio dell'applicazione demo -v4.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 planets-demo, con un prefisso chiave, o cartella, denominato v1 e un archivio denominato. `planetsdemo-v4.zip`



> **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": "planets-demo",
            "s3-key-prefix": "v1"
        }
    }],
    "definition": {
        "listeners": [{
            "port": 8196,
            "type": "http"
        }],
        "ba-application": {
            "app-location": "${s3-source}/PlanetsDemo-v4.zip"
        }
    }
}
```

Modificate la `s3-bucket` voce inserendo il nome del file zip dell'applicazione di esempio (ad es.`planets-demo`) e la `app-location` voce nel percorso S3 in cui è stato memorizzato il file zip dell'applicazione di esempio (ad es.`${s3-source}/PlanetsDemo-v4.zip`).

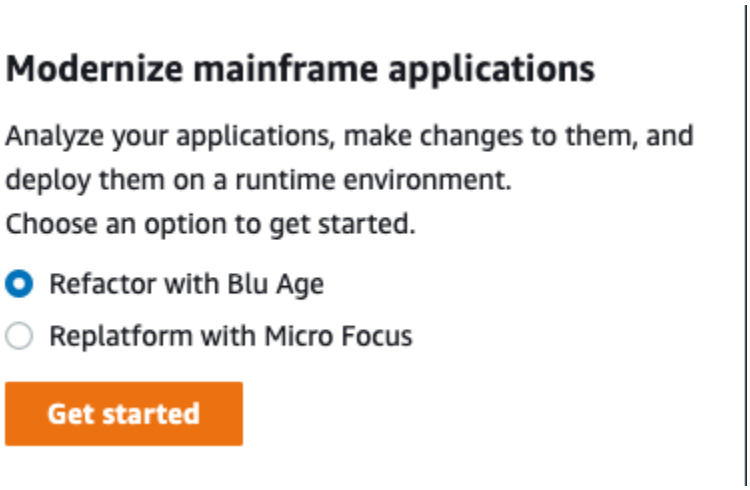
> **Note**
>
> Assicurati di creare il file di definizione dell'applicazione sul tuo locale come file di testo.

Per ulteriori informazioni sulla definizione dell'applicazione, vedereAWS Esempio di definizione dell'applicazione Transform for mainframe.

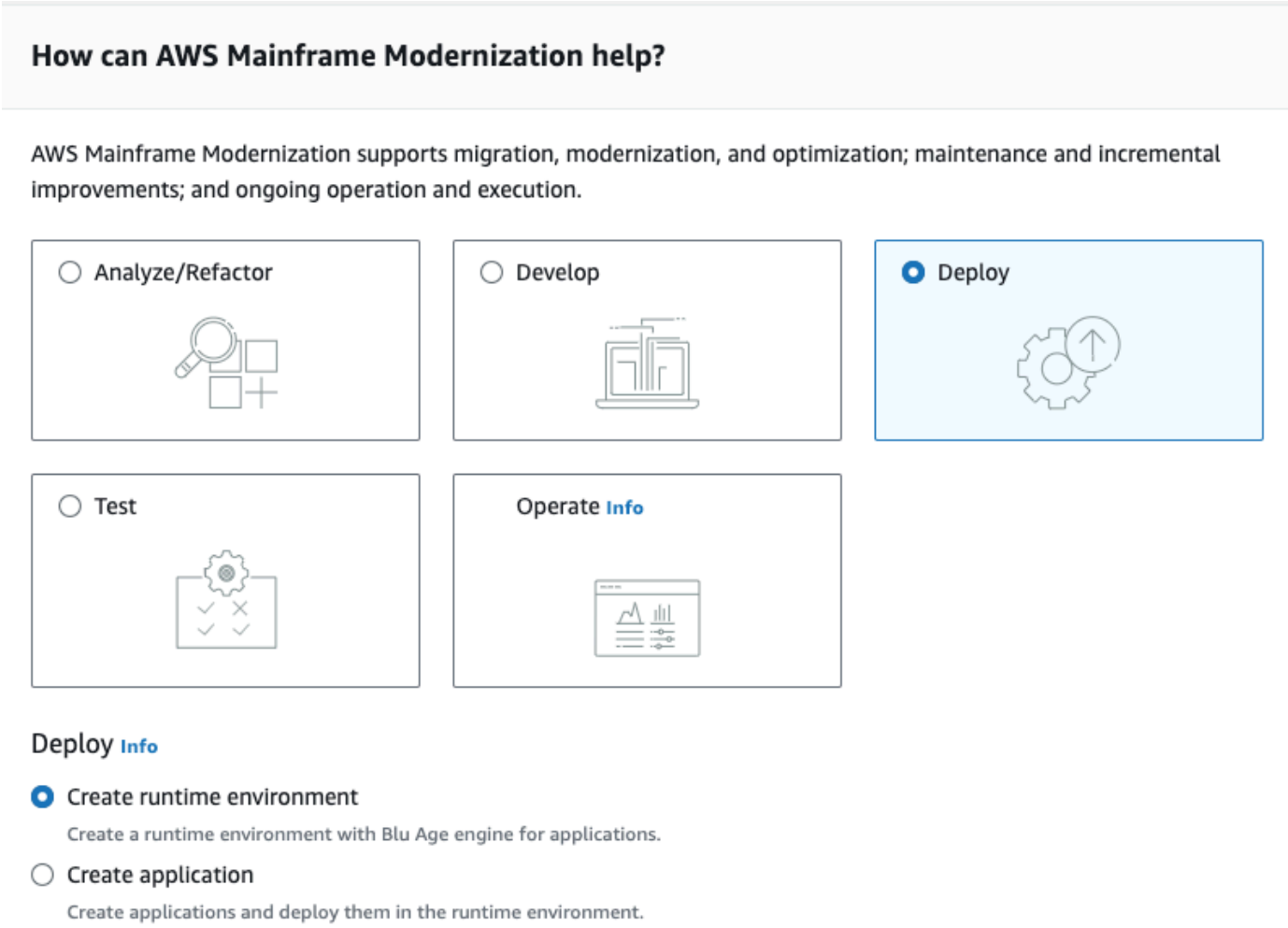
Fase 3: Creare un ambiente di runtime

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

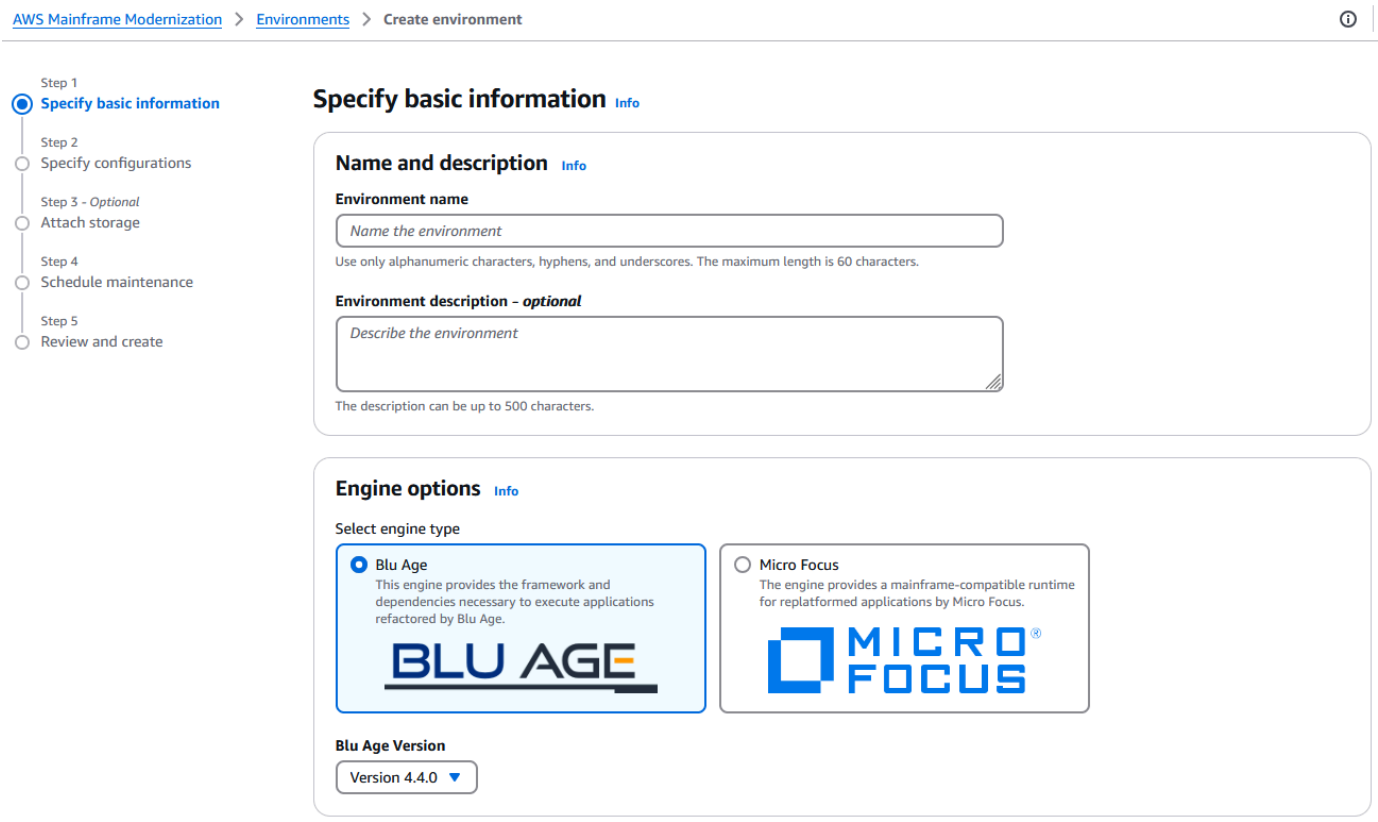
1. Apri la console di modernizzazione del AWS mainframe.
2. Nel Regione AWS selettore, scegliete la regione in cui desiderate creare l'ambiente. Regione AWS Deve corrispondere alla regione in cui hai creato il bucket S3. Passaggio 1: carica l'applicazione demo
3. In Modernizza le applicazioni mainframe, scegli Refactor with AWS Transform for mainframe, quindi scegli Inizia.



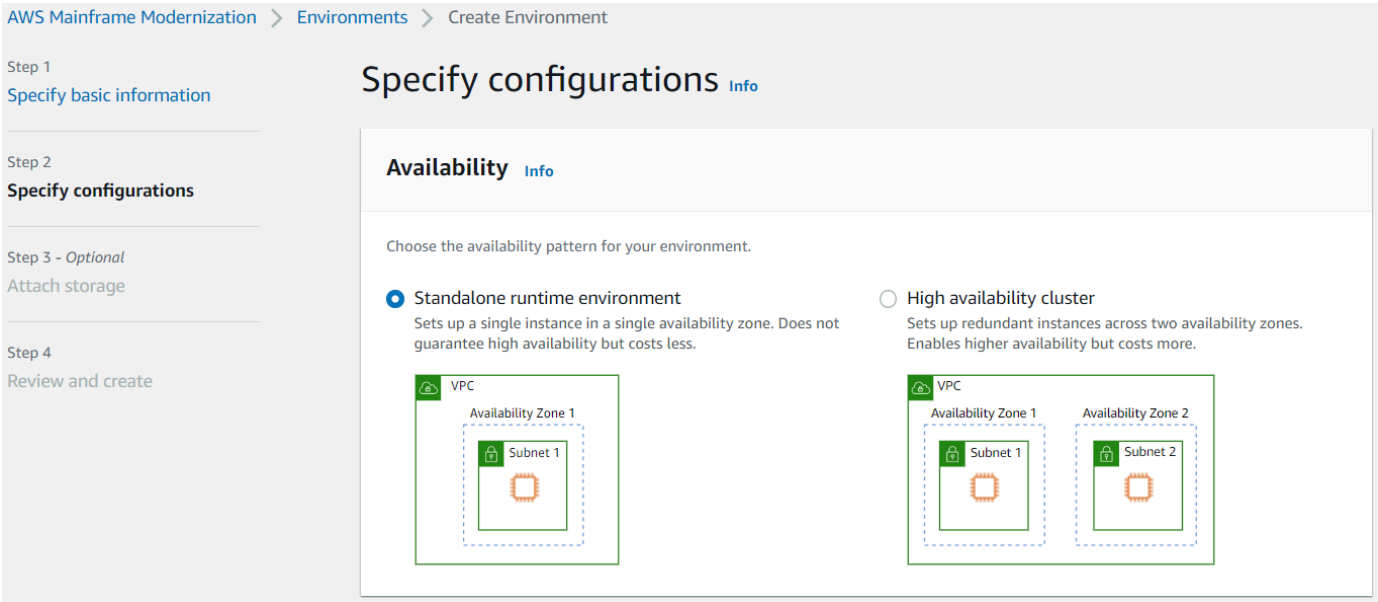
4. In Come può aiutare AWS Mainframe Modernization, scegli Deploy and Create runtime environment.



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.



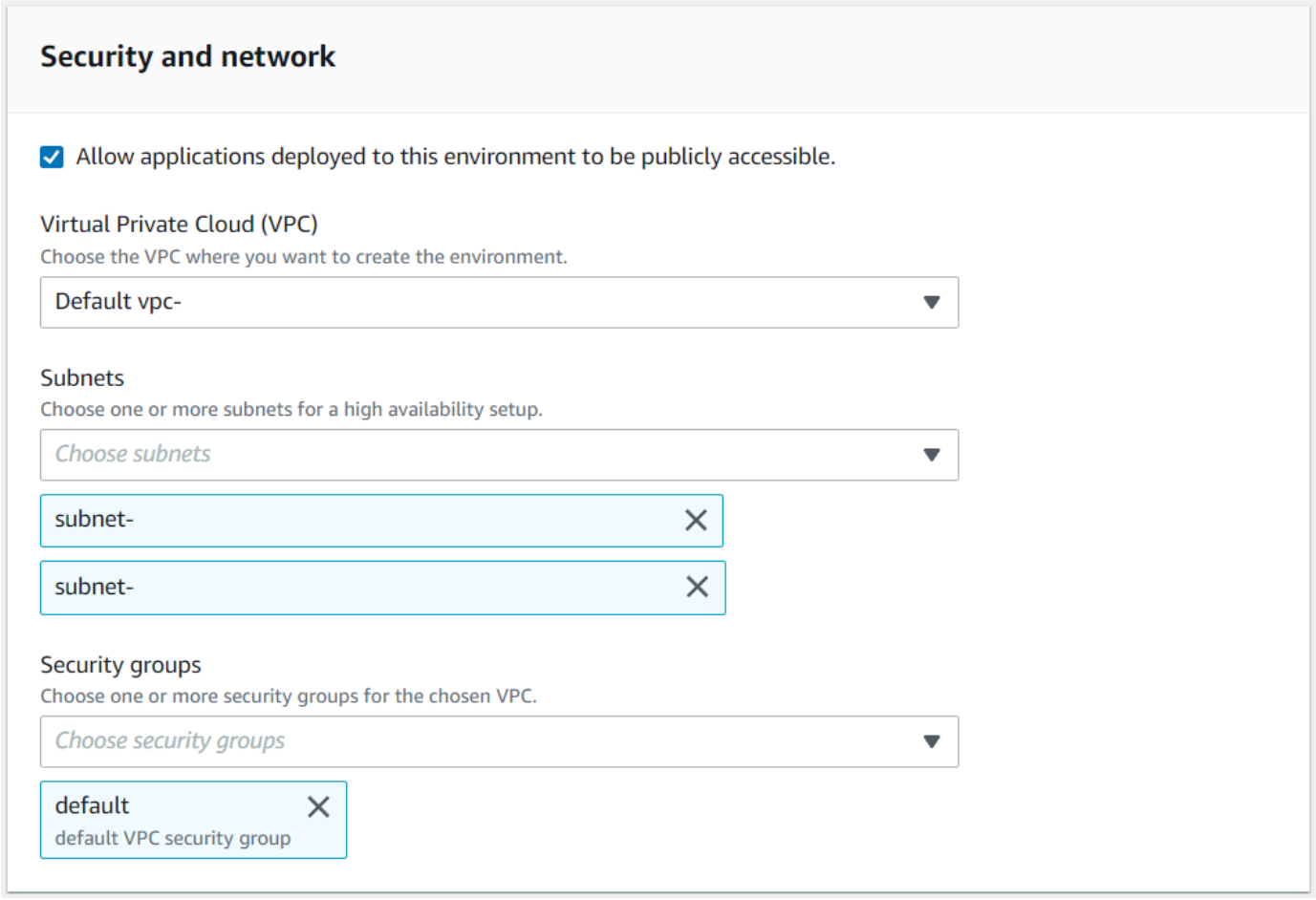
6. Nella pagina Specificare le configurazioni, selezionare Ambiente di runtime autonomo.



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

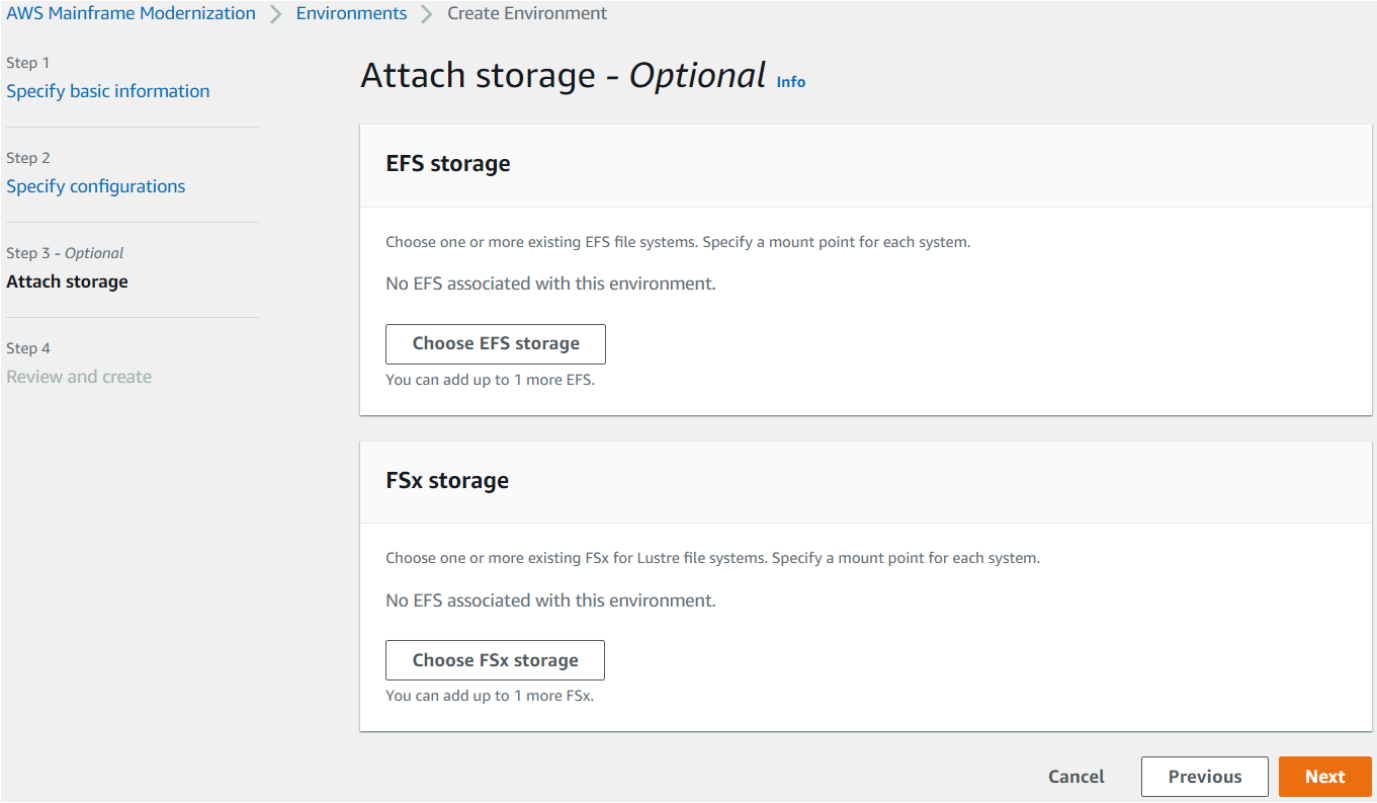
   - Scegli Consenti alle applicazioni distribuite in questo ambiente di essere accessibili al pubblico. Questa opzione assegna un indirizzo IP pubblico all'applicazione in modo che sia possibile accedervi dal desktop.

- 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.



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 Errore di risoluzione dei problemi: impossibile accedere all'URL di un'applicazione.

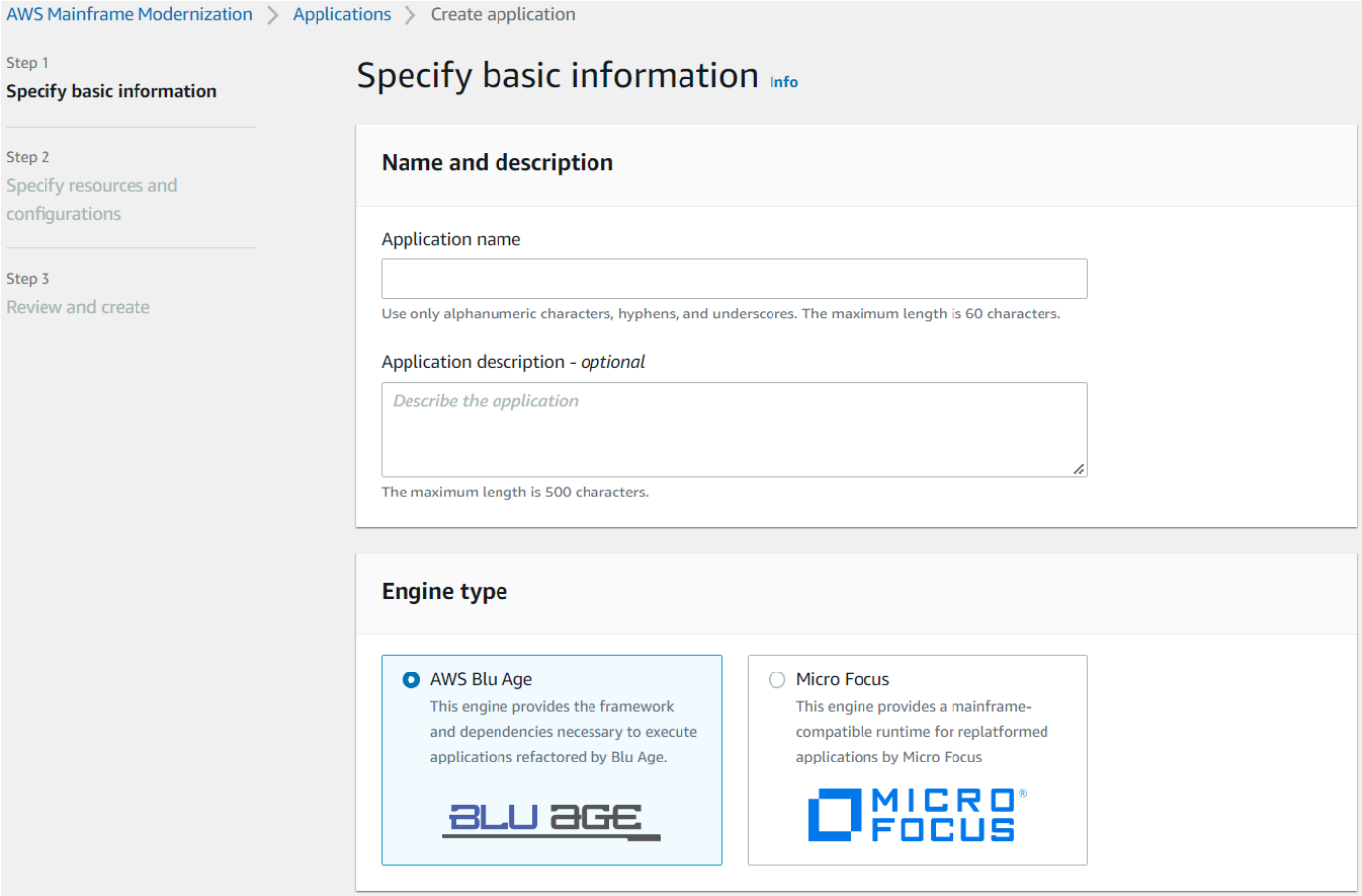
8. Scegli Next (Successivo).
9. In Allega archiviazione - Facoltativo, lascia le selezioni predefinite e scegli Avanti.



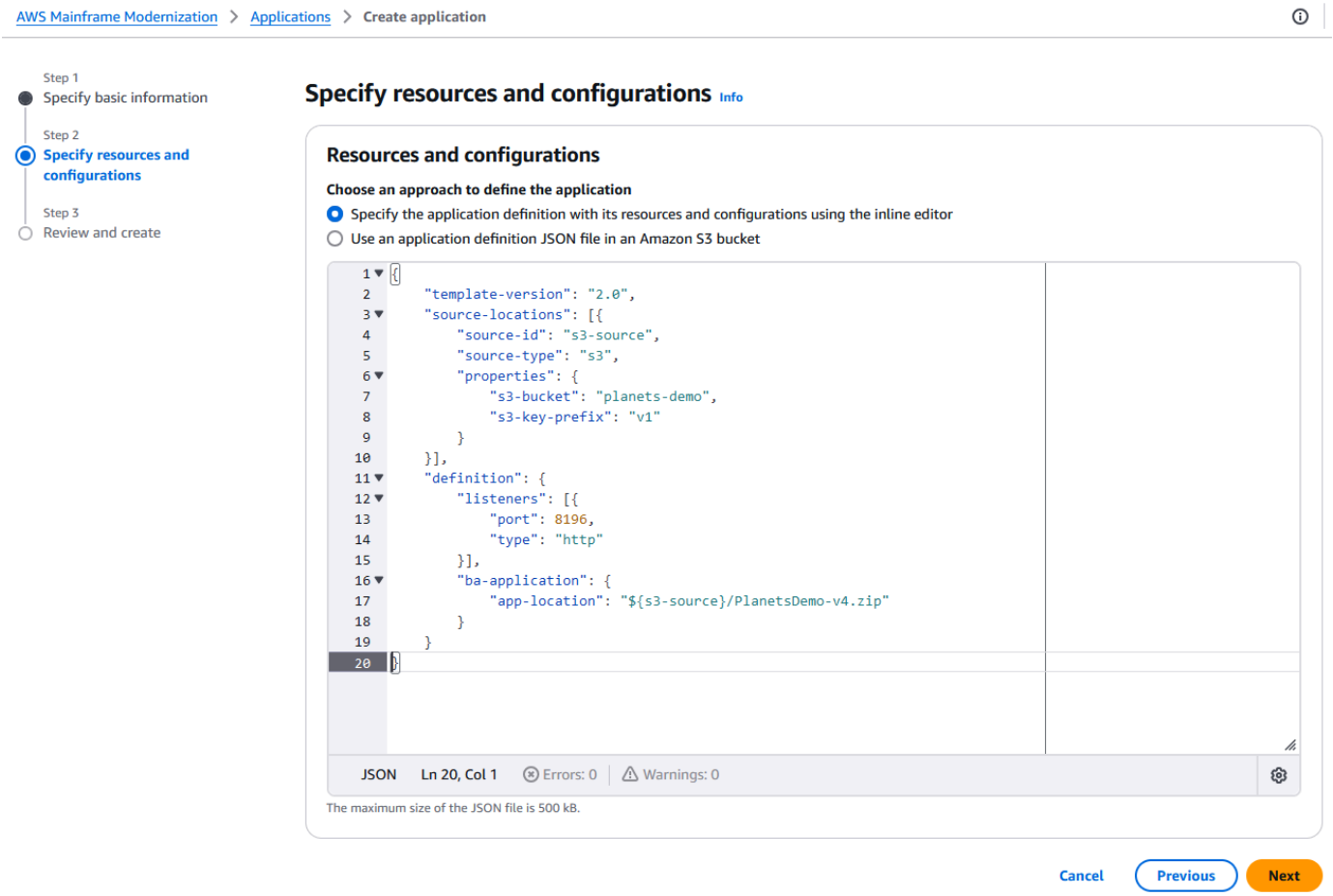
10. In Pianifica manutenzione, scegli Nessuna preferenza, quindi scegli Avanti.
11. In Rivedi e crea, esamina le informazioni, quindi scegli Crea ambiente.

Fase 4: creare un'applicazione

1. Passa a AWS Mainframe Modernization in. Console di gestione AWS
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 AWS Transform for mainframe. Quindi scegli Successivo.



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"



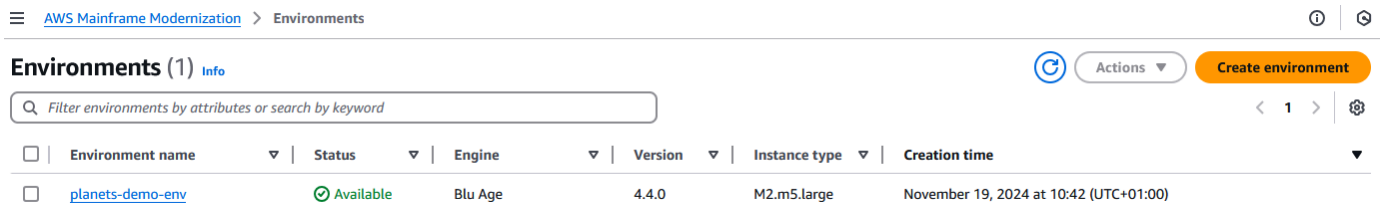
4. In Rivedi e crea, esamina le tue scelte, quindi scegli Crea applicazione.

> **Note**
>
> Se la creazione dell'applicazione fallisce, controlla il percorso S3 che hai inserito poiché fa distinzione tra maiuscole e minuscole.

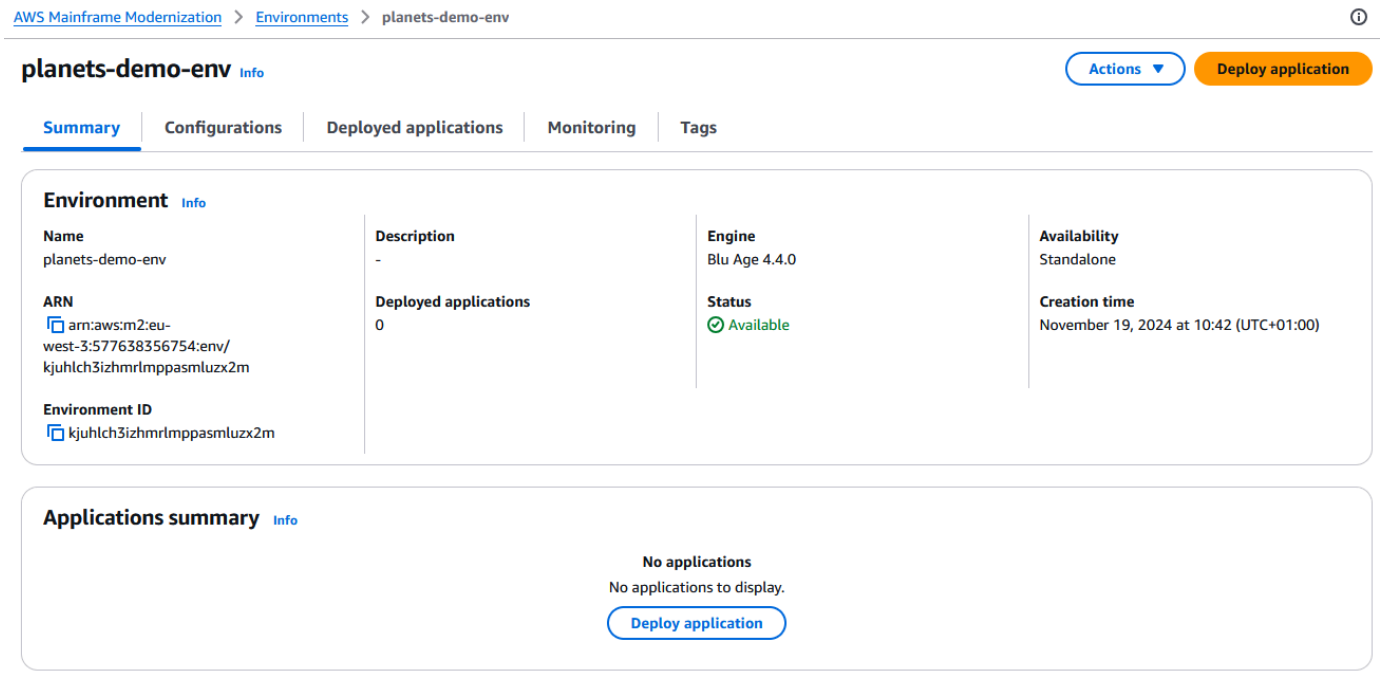
Fase 5: Implementazione di un'applicazione

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.

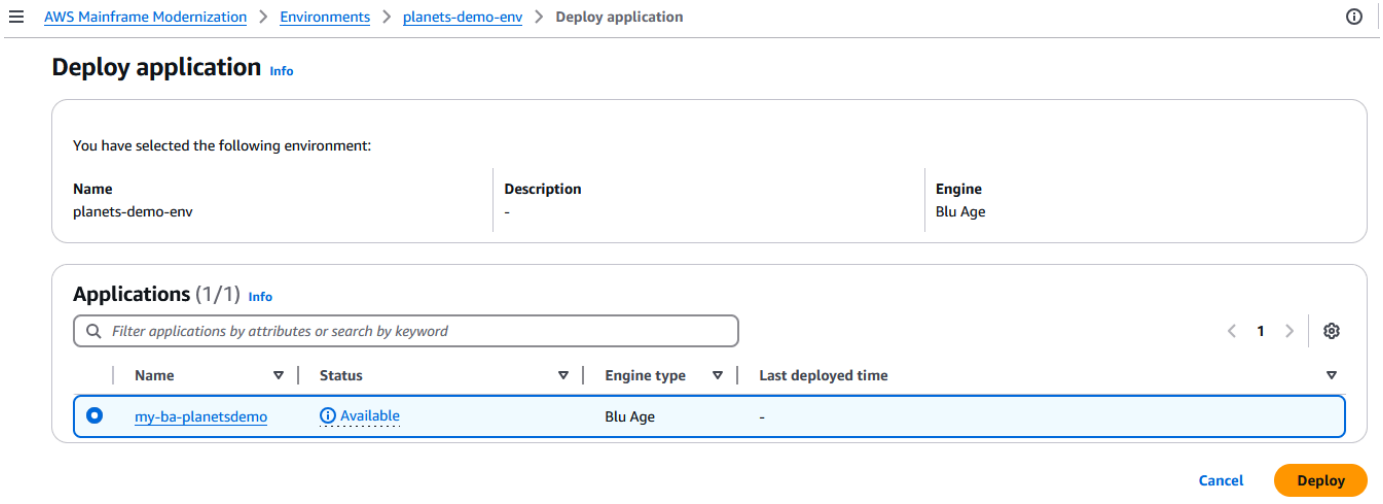
1. Passa alla modernizzazione di AWS Mainframe nella console di AWS gestione. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo). Viene visualizzata la pagina con l'elenco degli ambienti.



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



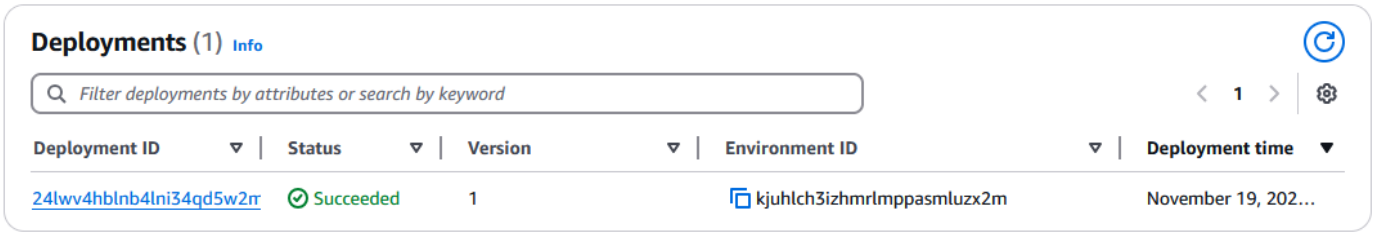
4. Scegli l'applicazione creata in precedenza, quindi scegli la versione in cui vuoi distribuire l'applicazione. Scegliere quindi Deploy (Distribuisci).



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 Console di gestione AWS e scegli Applicazioni.
2. Scegli la tua applicazione, quindi vai a Distribuzioni. Lo stato della domanda dovrebbe essere Riuscito.



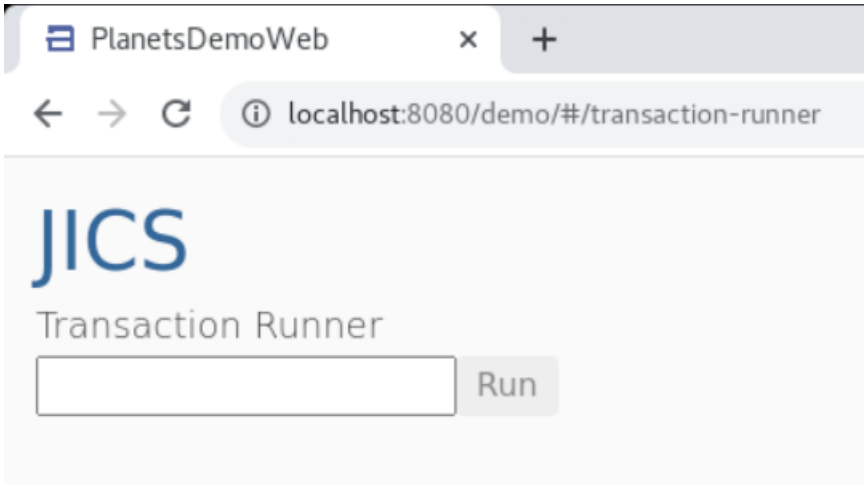
3. Scegli Azioni, quindi scegli Avvia applicazione.

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, accedi 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. Errore di risoluzione dei problemi: 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 proveniente da i/p LB sulla porta 8196.

Fase 8: test dell'applicazione

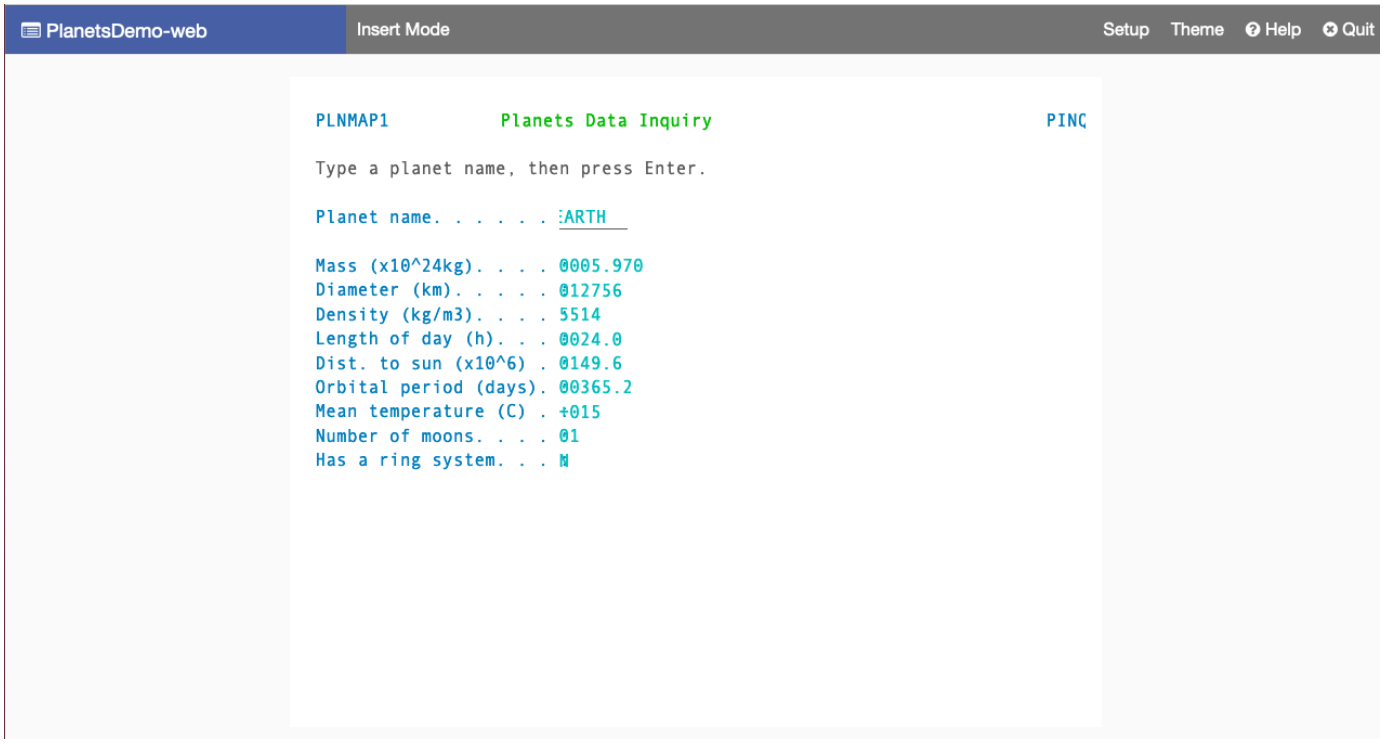
In questo passaggio, si esegue una transazione nell'applicazione migrata.

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.



Dovresti vedere i dettagli sul pianeta.

Eseguire la 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 Modernizzazione del mainframe AWS.
- Eliminare l'ambiente di runtime. Per ulteriori informazioni, consulta Eliminare un ambiente di runtime di AWS modernizzazione del mainframe.

Tutorial: Configurazione del runtime gestito per Rocket Software (precedentemente Micro Focus)

È possibile distribuire ed eseguire un'applicazione in un ambiente di runtime gestito da AWS Mainframe Modernization con il motore di runtime Rocket Software. 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 Rocket Software. 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. Gestione dei segreti AWS

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 Gestione dei segreti AWS database segreto
- Passaggio 5: aggiungere SSLMode al segreto
- Fase 6: Creare un ambiente di runtime
- Fase 7: Creare un'applicazione
- Fase 8: Implementazione di un'applicazione

- Fase 9: Importazione dei set di dati
- Fase 10: Avviare un'applicazione
- Fase 11: Connect all'applicazione CardDemo CICS
- Eseguire la pulizia delle risorse
- Fasi successive

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, puoi avviare un'istanza del software AWS Mainframe Modernization WorkSpaces Applications Rocket e utilizzare l'emulatore Rumba 3270 (non disponibile gratuitamente).

  Per informazioni sulle applicazioni, vedere. WorkSpaces the section called "Tutorial: Configurazione WorkSpaces delle applicazioni 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 decomprimete il file scaricato in qualsiasi directory locale. Questa directory conterrà una sottodirectory intitolata. `CardDemo_runtime`
- 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

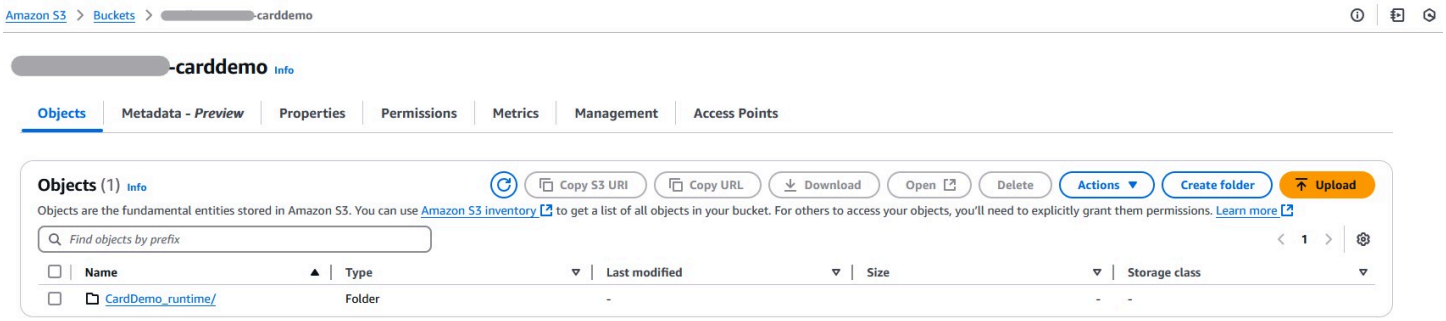
In questo passaggio, crei un bucket Amazon S3 e carichi CardDemo i file in questo bucket. Più avanti in questo tutorial, utilizzerai questi file per distribuire ed eseguire l'applicazione di CardDemo esempio in un ambiente AWS Mainframe Modernization Rocket Software Managed Runtime.

> **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 Rocket Software 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_runtime` 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_runtime`



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. Gestione dei segreti AWS Inserisci invece una password principale e confermala.

> **Note**
>
> Salva il nome utente e la password che usi per il database. Li archivierai 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 Create (Crea).

Per configurare il gruppo di parametri personalizzato

1. Scegliete il gruppo di parametri appena creato.
2. Seleziona Azioni, quindi scegli 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. Scegliete Continua e controllate 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 prima un. AWS KMS key

Per creare un AWS KMS key

1. Aprire 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. Scegli Next (Successivo).
5. Assegna alla chiave un alias come e una descrizione opzionale. `MicroFocus-Tutorial-RDS-Key`
6. Scegli Next (Successivo).
7. Assegna un amministratore chiave selezionando la casella accanto al tuo utente o ruolo.
8. Scegli Next (Successivo).
9. Assegna l'autorizzazione all'utilizzo delle chiavi selezionando la casella accanto al tuo utente o ruolo.
10. Scegli Next (Successivo).
11. Nella schermata di revisione, modifica la politica chiave, quindi inserisci quanto segue all'interno dell'array «Statement» esistente:

```
{
     "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.

12. 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 Gestione dei segreti AWS database segreto

Ora archivia le credenziali del database in modo sicuro utilizzando and. Gestione dei segreti AWS AWS KMS key

Per creare e configurare un database segreto Gestione dei segreti AWS

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:

JSON

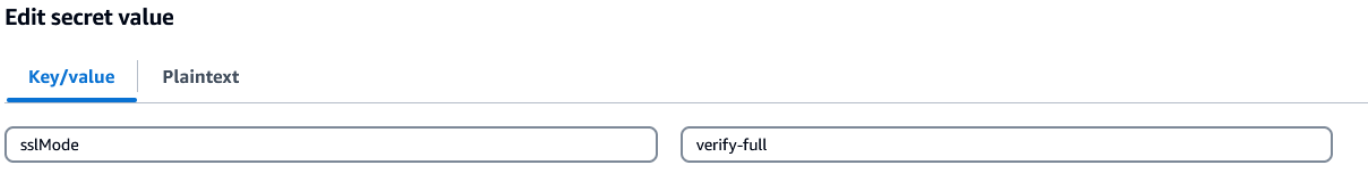
```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect" : "Allow",
            "Principal" : {
                "Service" : "m2.amazonaws.com"
            },
            "Action" : "secretsmanager:GetSecretValue",
            "Resource" : "*"
        }
    ]
}
```

10. Scegli Save (Salva).
11. Scegli Avanti per le schermate successive, quindi scegli Store.

Passaggio 5: aggiungere SSLMode al segreto

Per aggiungere SSLMode al segreto

1. Aggiorna l'elenco dei segreti per visualizzare il nuovo segreto.
2. Scegli il segreto appena creato nel passaggio 4 e nota `Secret ARN` perché ti servirà più avanti nel tutorial.
3. Nella scheda Panoramica del segreto, scegli Recupera valore segreto.
4. Scegli Modifica, quindi scegli Aggiungi riga.
5. Aggiungi una chiave per `sslMode` con un valore di`verify-full`:

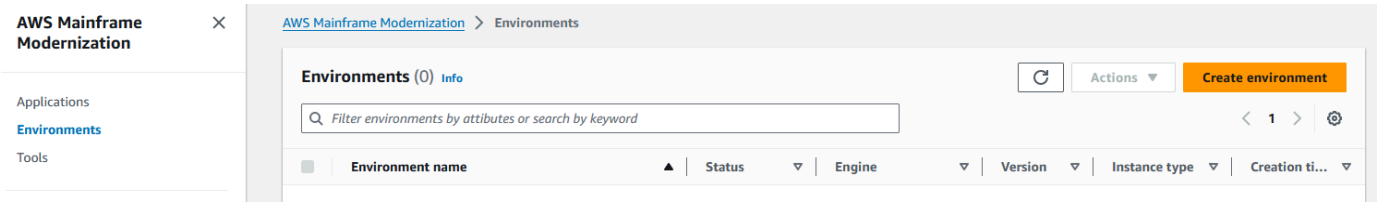


6. Scegli Save (Salva).

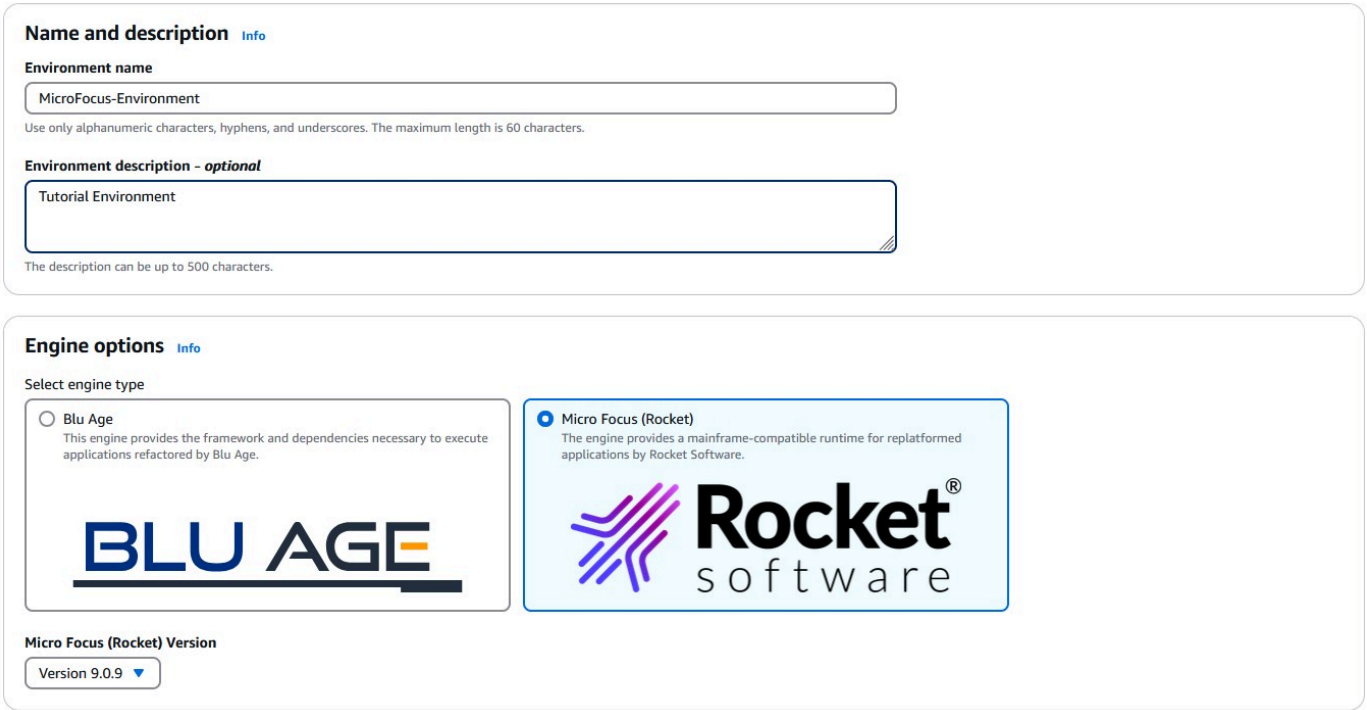
Fase 6: Creare un ambiente di runtime

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 (Rocket) sia selezionato.
   c. Scegliete la versione più recente di Micro Focus (Rocket).
   d. Scegli Next (Successivo).



4. Configura l'ambiente

a. In Disponibilità, scegli Cluster ad alta disponibilità.

b. In Risorse, scegli 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. Scegli Next (Successivo).

Specify configurations Info

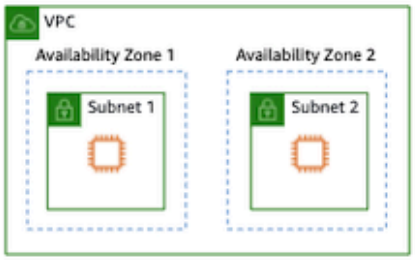
Availability Info

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.

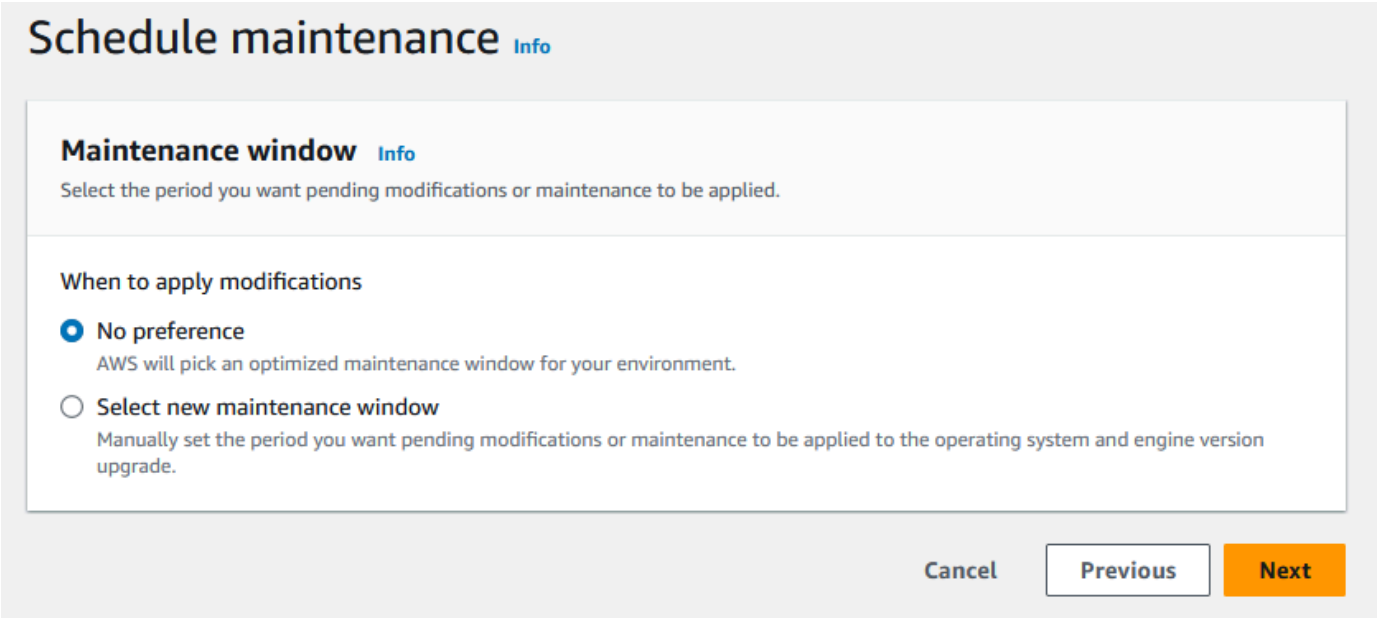
Choose subnets

subnet-56f1a | us-west-2a

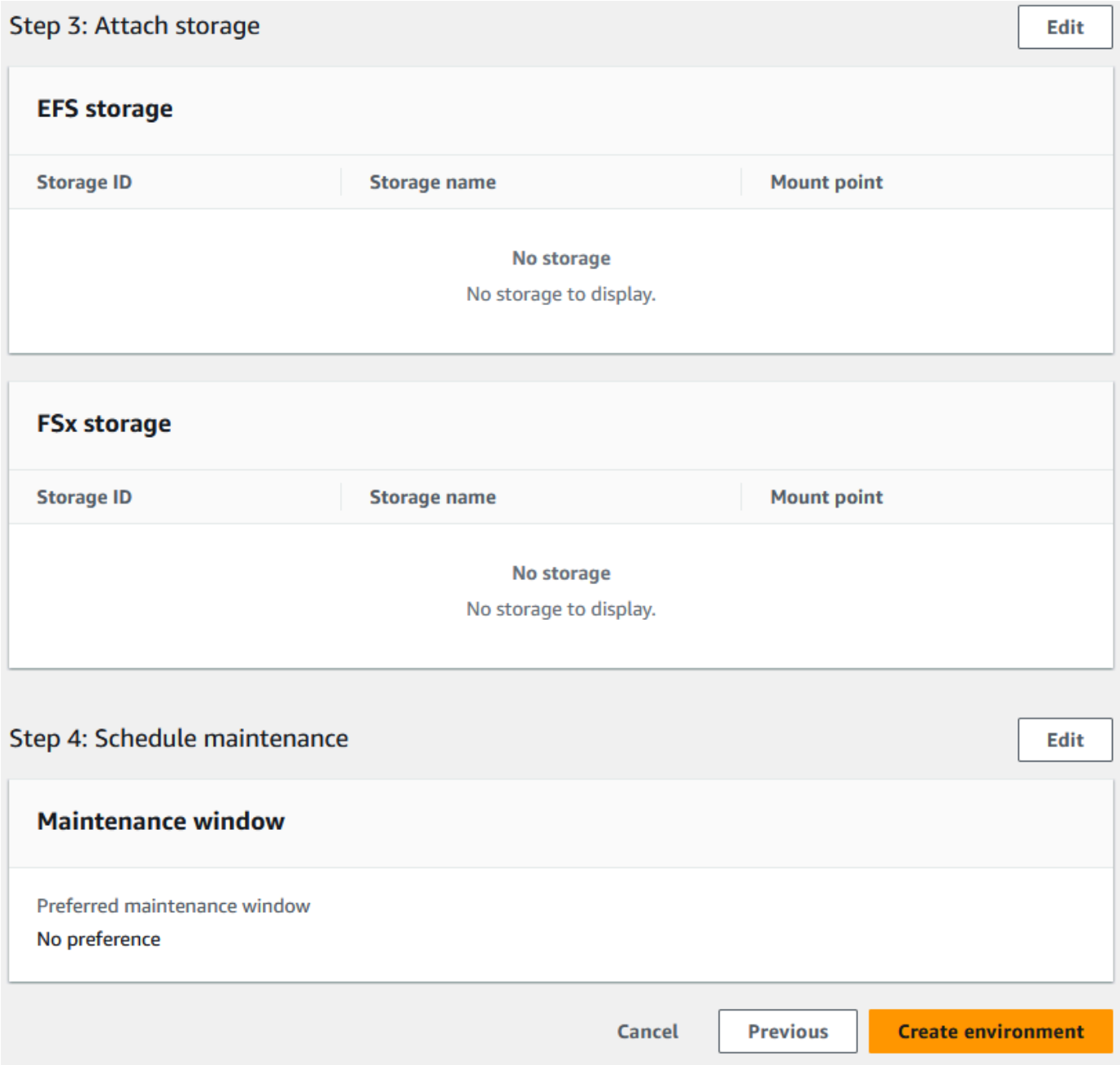
subnet-6685 | us-west-2b

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

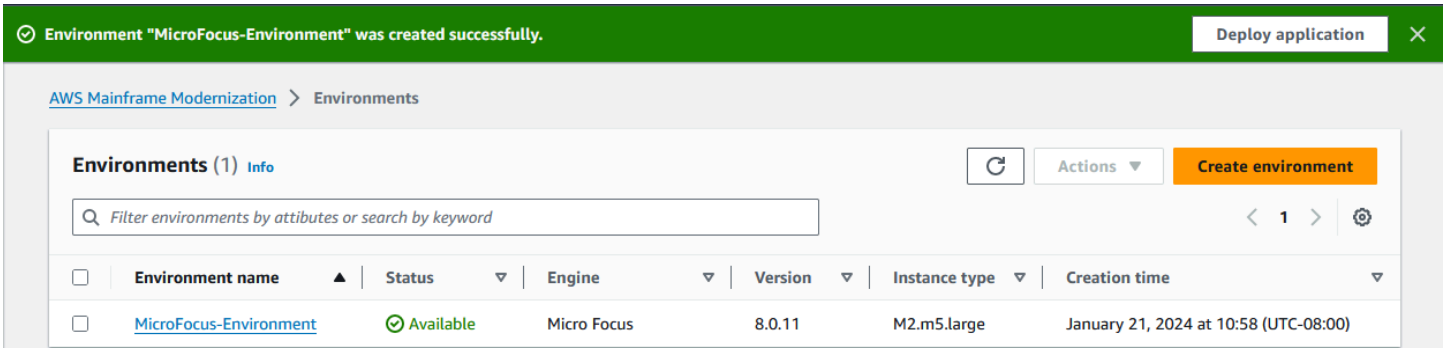
5. Nella pagina Allega archiviazione, scegli Avanti.
6. Nella pagina Pianifica manutenzione, scegli Nessuna preferenza, quindi scegli Avanti.



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



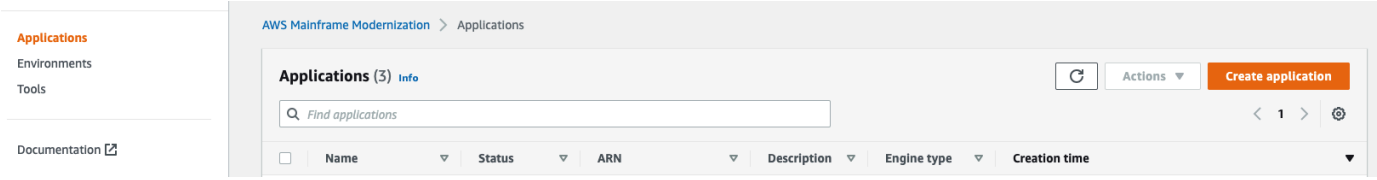
Dopo aver creato l'ambiente, viene visualizzato un banner che dice`Environment` *`name`* `was created successfully`, e il campo Stato diventa Disponibile. Il processo di creazione dell'ambiente richiede alcuni minuti, ma puoi continuare con i passaggi successivi mentre è in esecuzione.



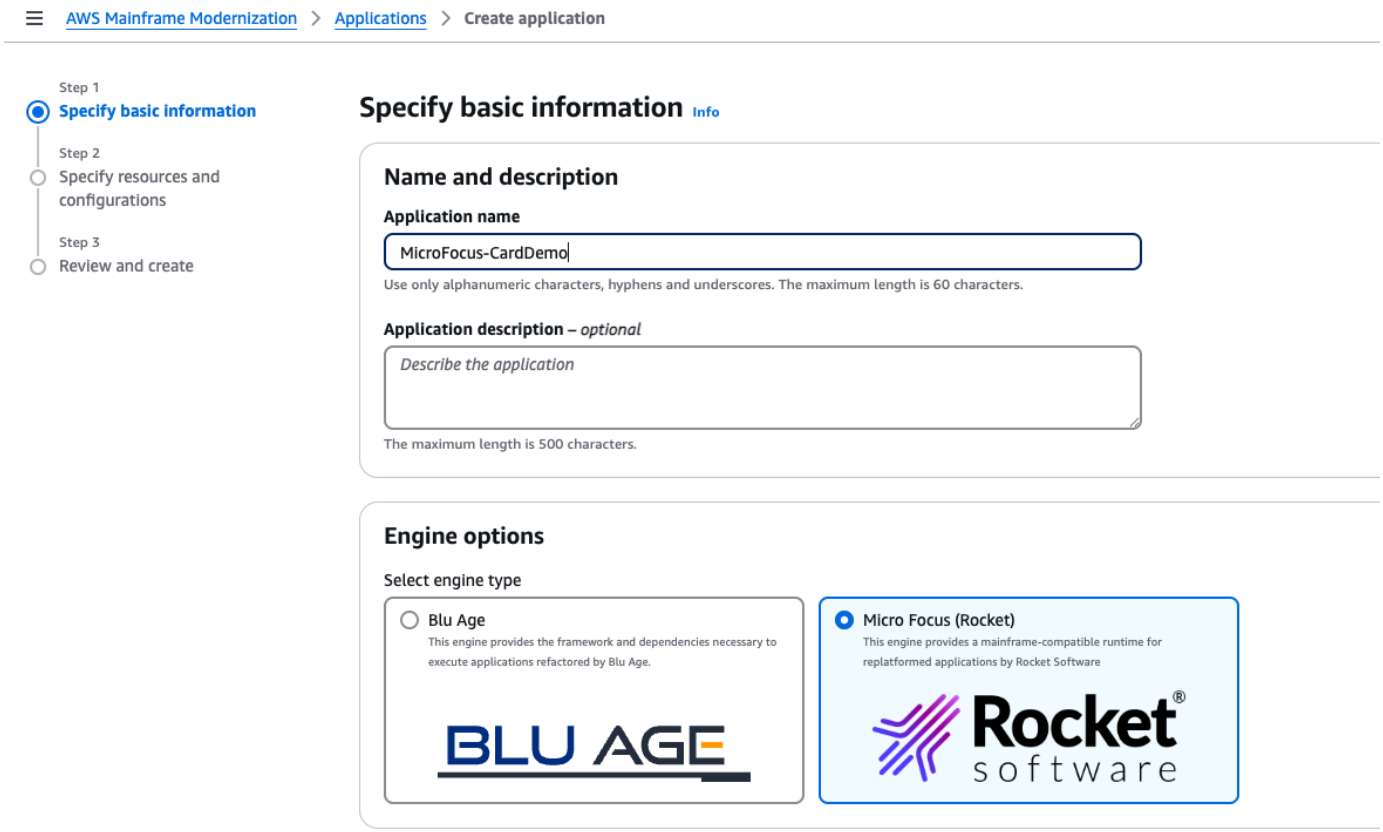
Fase 7: Creare un'applicazione

Come creare un'applicazione

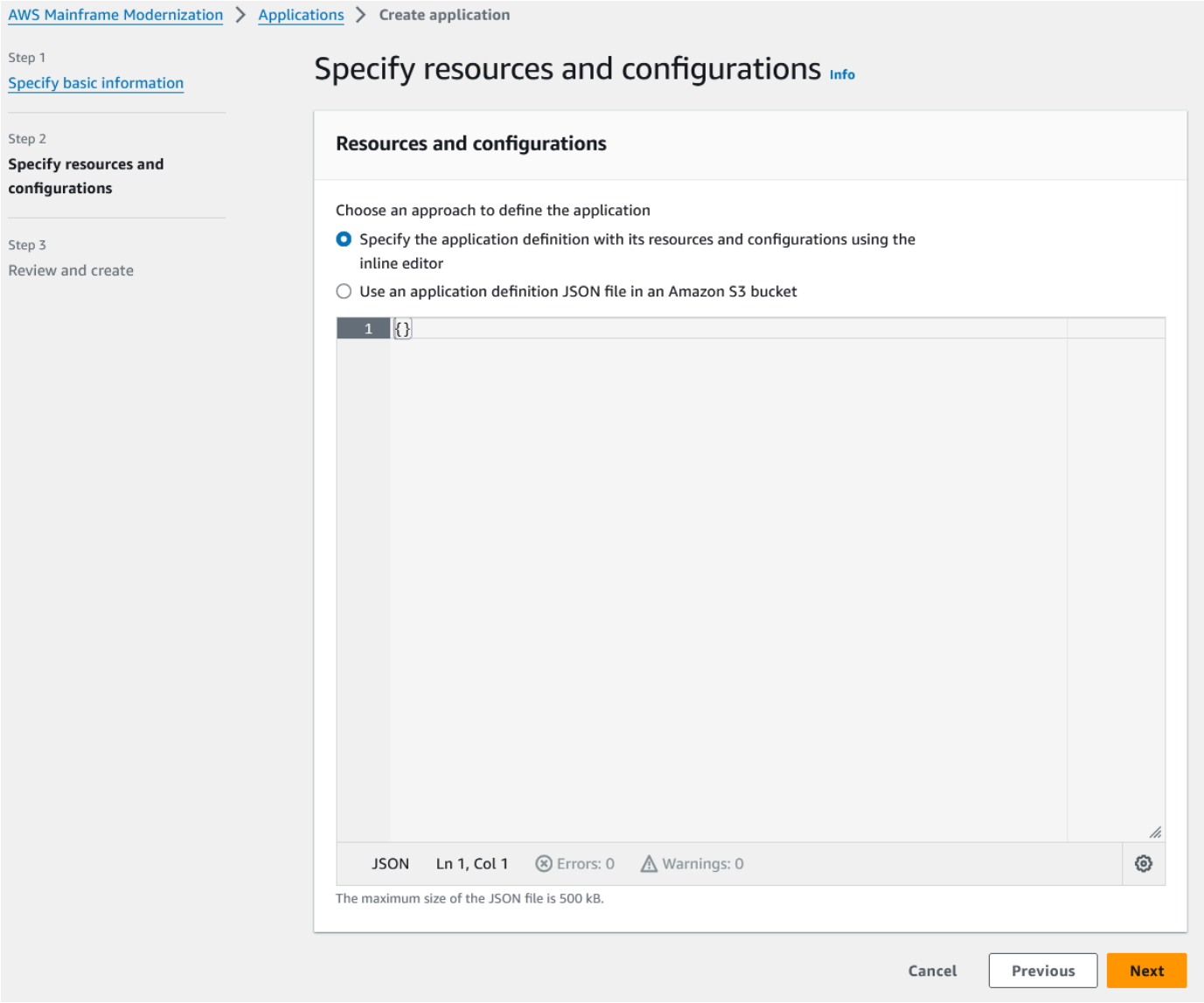
1. Nel riquadro di navigazione, scegliere Applications (Applicazioni). Quindi scegli Crea applicazione.



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 (Rocket) sia selezionato. Quindi scegli Successivo.



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



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_runtime"
      }
    }
```

```
  ],
  "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"
      }
    ]
  }
}
```

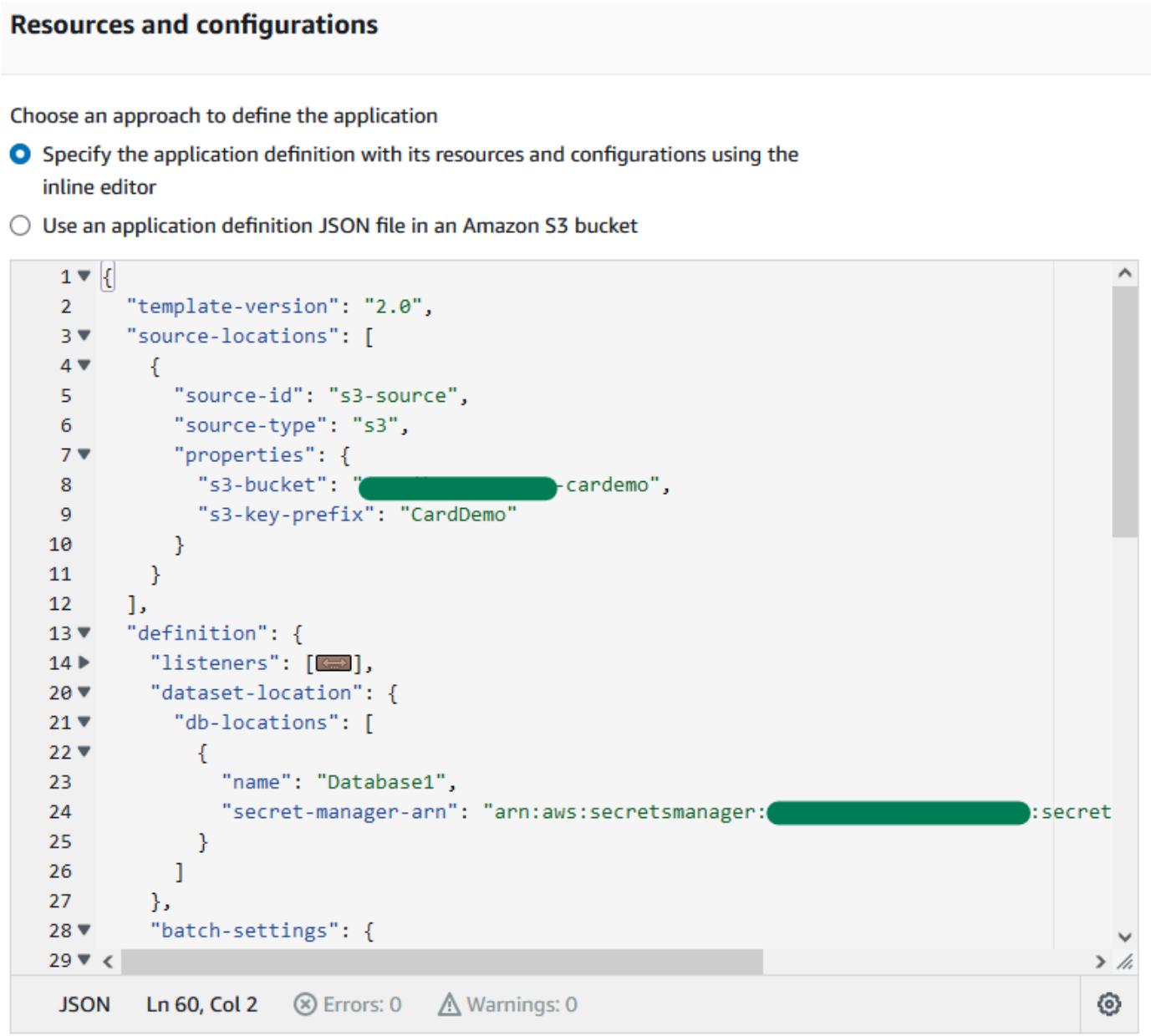
> **Note**
>
> Questo file è soggetto a modifiche.

4. Modifica l'applicazione JSON 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.

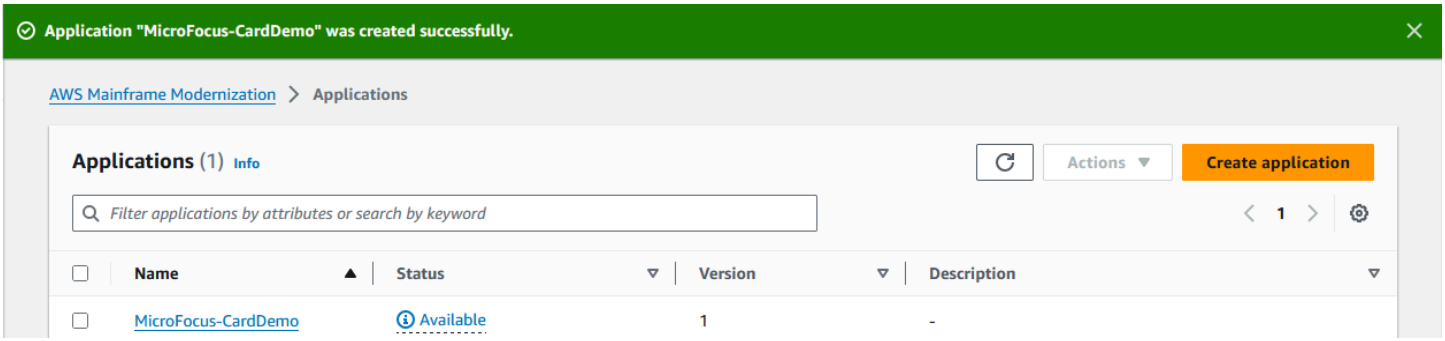
   b. Sostituisci il valore per `s3-key-prefix` con la cartella (key prefix) in cui hai caricato i file di CardDemo esempio. Se hai caricato la `CardDemo` directory direttamente in un bucket Amazon S3, `s3-key-prefix` non è necessario modificarla.

   c. Sostituisci entrambi `secret-manager-arn` i valori con l'ARN per il segreto del database creato nel passaggio 4.



Per ulteriori informazioni sulla definizione dell'applicazione, vedereDefinizione dell'applicazione Rocket Software (precedentemente Micro Focus).

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

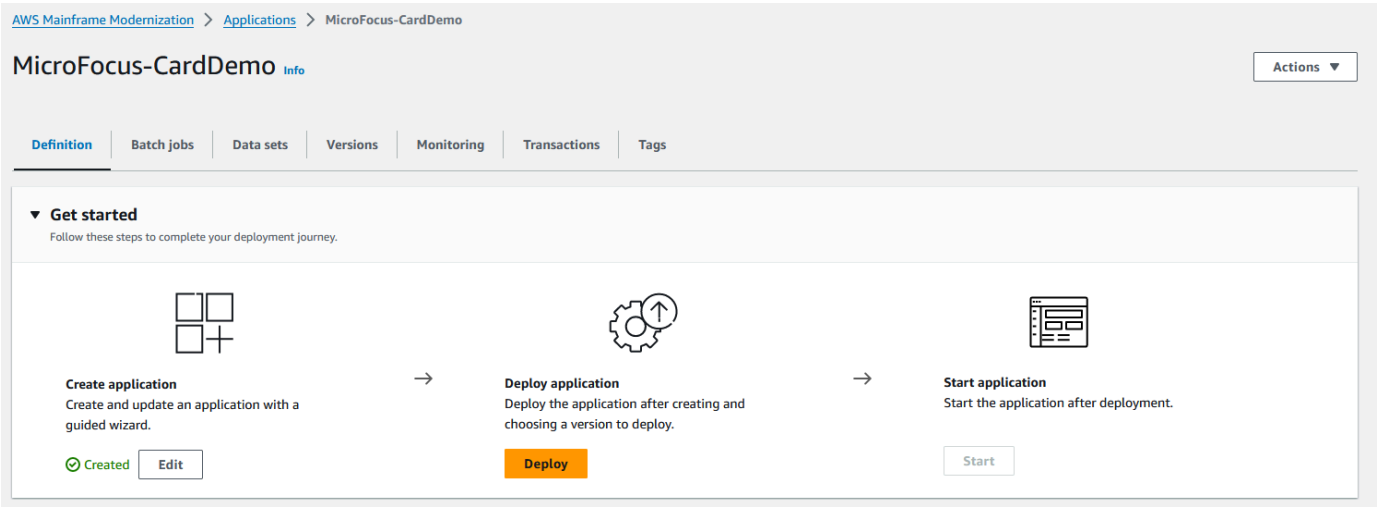


Dopo aver creato l'applicazione, viene visualizzato un banner che dice`Application` *`name`* `was created successfully`. E il campo Stato diventa Disponibile.

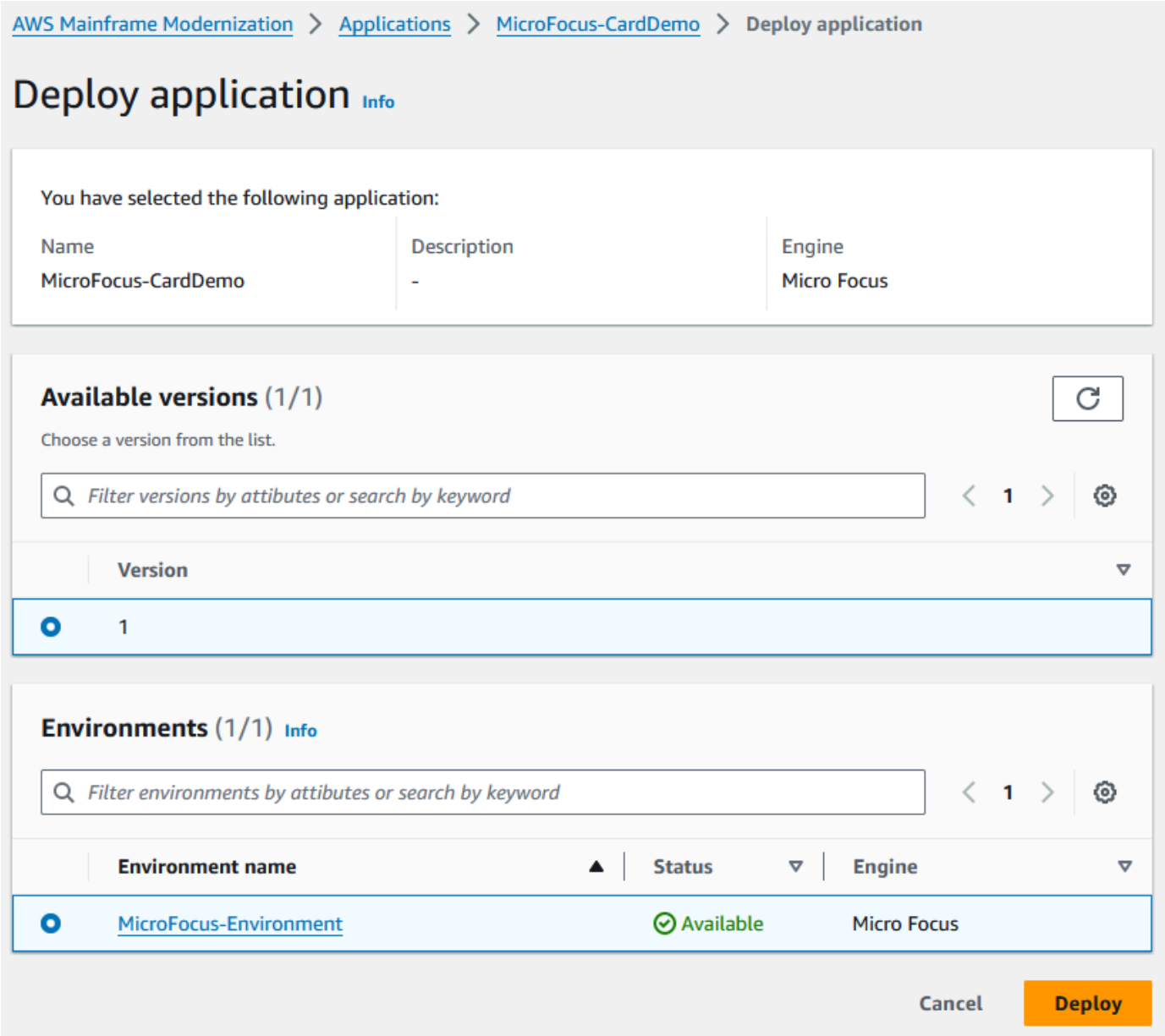
Fase 8: Implementazione di un'applicazione

Per distribuire un'applicazione

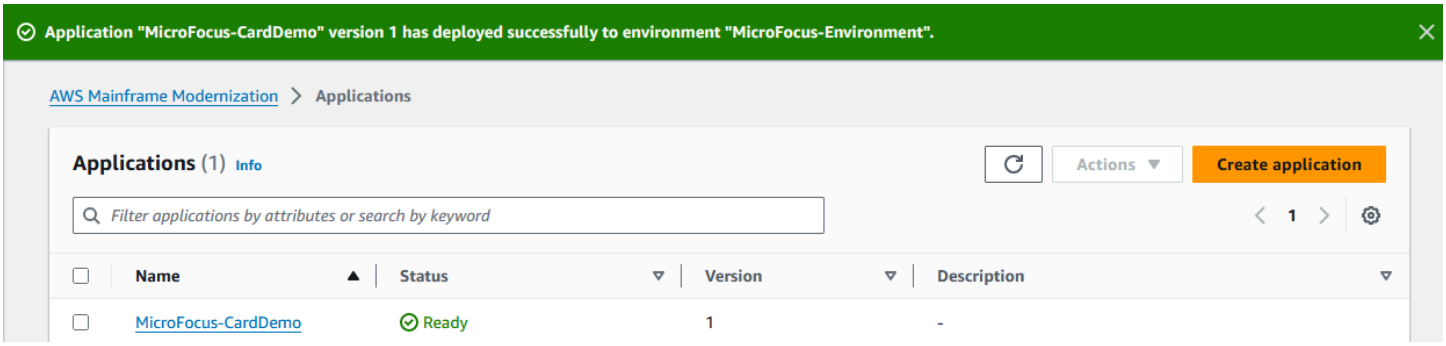
1. Nel riquadro di navigazione, scegli Applicazioni, quindi scegli`MicroFocus-CardDemo`.
2. In Distribuisci applicazione, scegli Distribuisci.



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



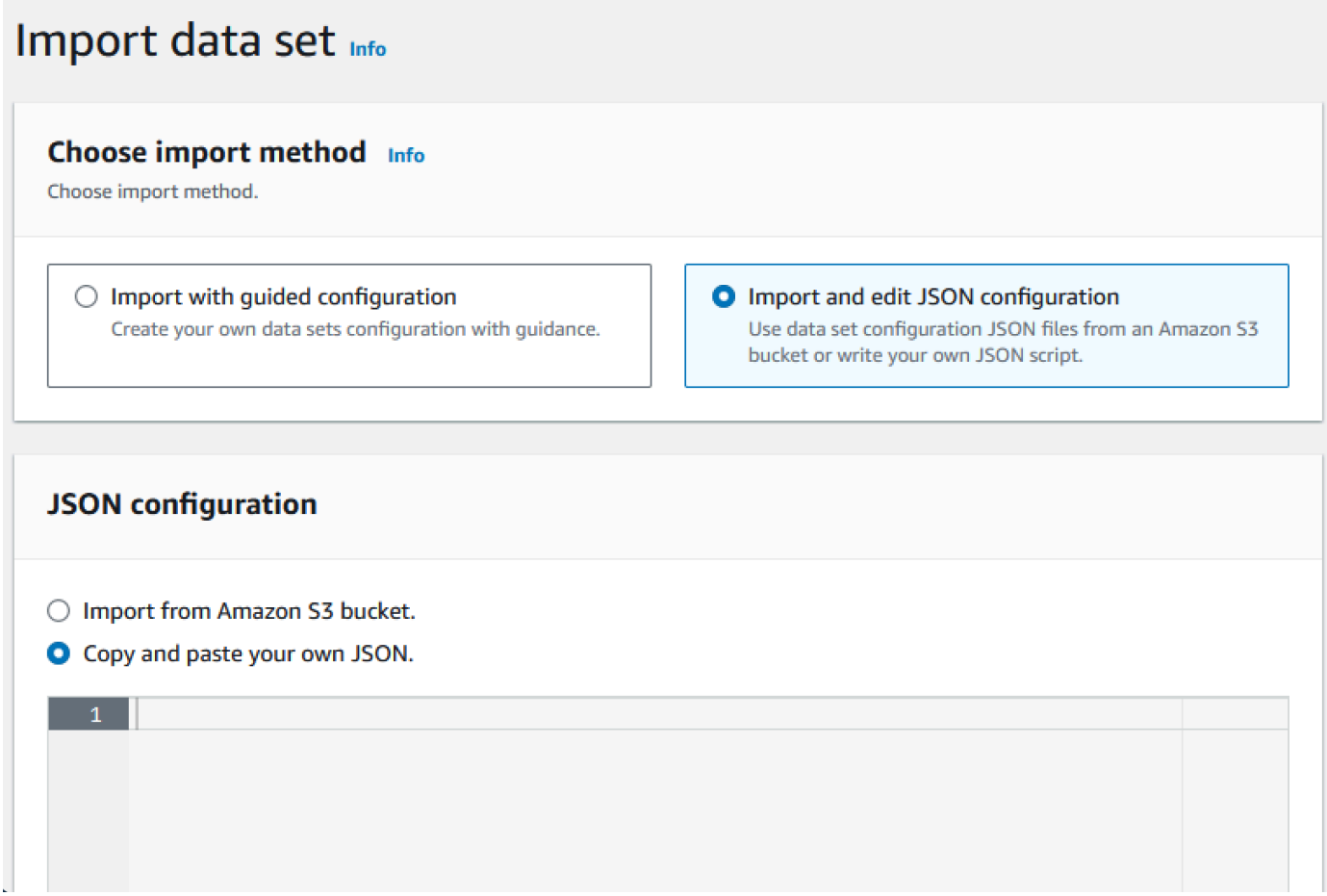
Quando l' CardDemo applicazione viene distribuita correttamente, lo stato diventa Pronto.



Fase 9: 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.



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_runtime/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_runtime/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_runtime/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_runtime/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_runtime/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_runtime/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_runtime/catalog/data/
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT"
            }
        },
        {
            "dataSet": {
                "storageType": "Database",
                "datasetName": "AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS",
                "relativePath": "DATA",
                "datasetOrg": {
```

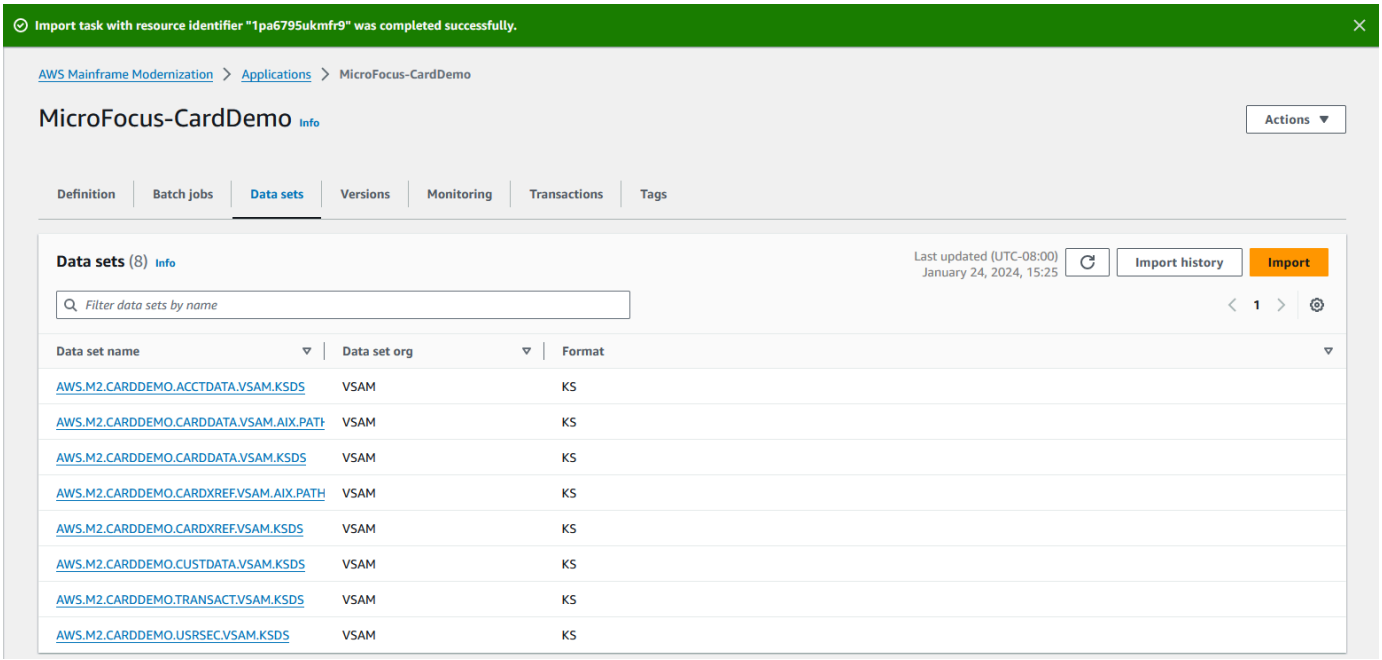
```
                    "vsam": {
                        "format": "KS",
                        "encoding": "A",
                        "primaryKey": {
                            "length": 8,
                            "offset": 0
                        }
                    }
                },
                "recordLength": {
                    "min": 80,
                    "max": 80
                }
            },
            "externalLocation": {
                "s3Location": "s3://<s3-bucket-name>/CardDemo_runtime/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.

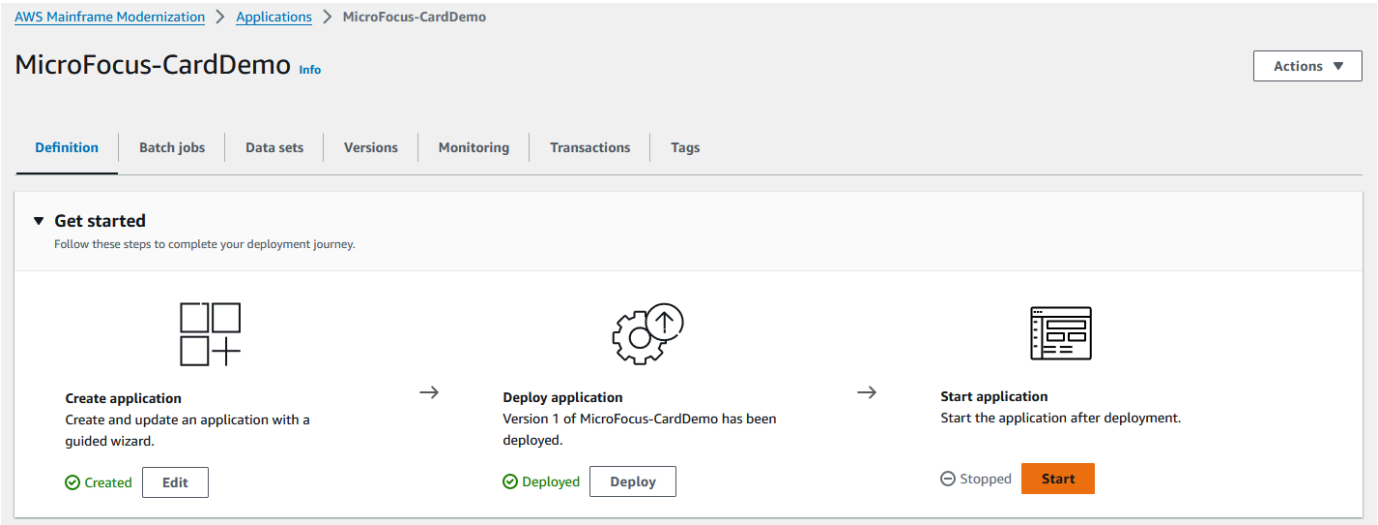


Puoi anche visualizzare lo stato di tutte le importazioni di set di dati scegliendo Cronologia importazioni nella scheda Set di dati.

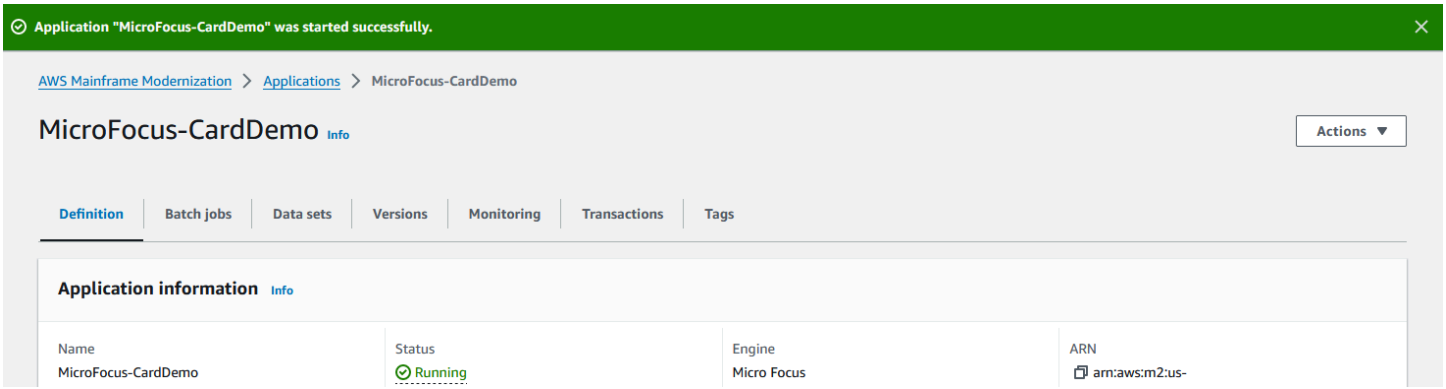
Fase 10: Avviare un'applicazione

Per avviare un'applicazione

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



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 11: 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 connetterai.

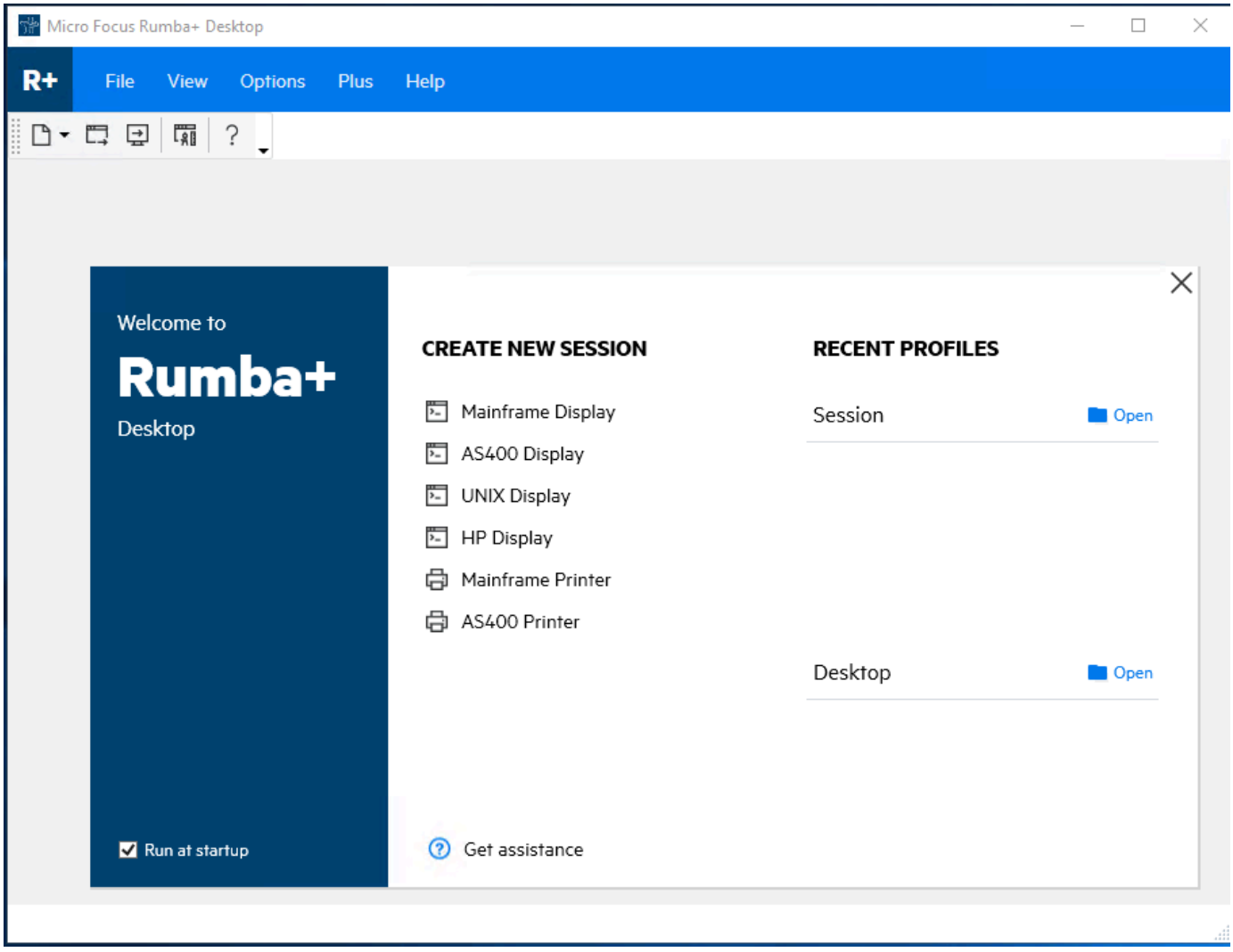
Per configurare la TN3270 connessione, sono necessari anche il nome host DNS e la porta dell'applicazione.

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

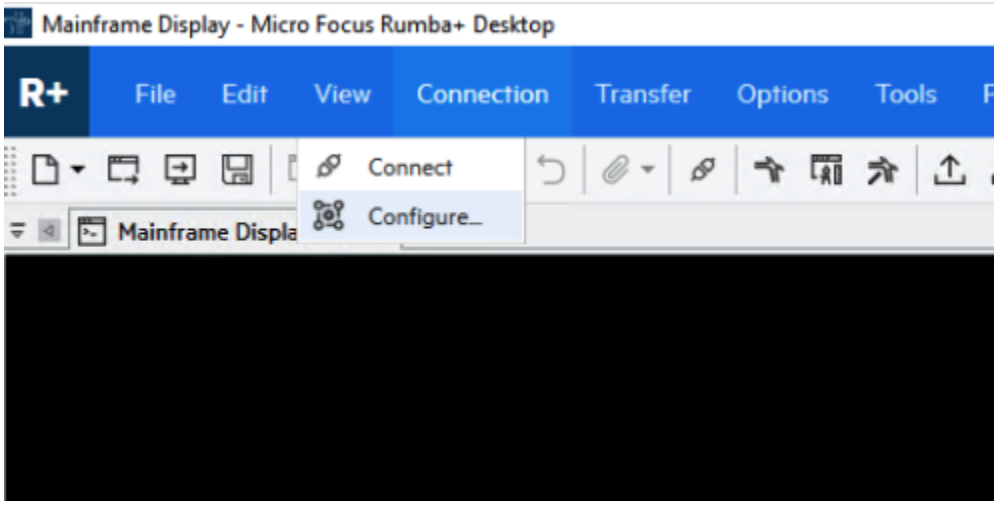
1. Apri la console di modernizzazione del AWS mainframe e scegli Applicazioni, quindi scegli. `MicroFocus-CardDemo`
2. Scegli l'icona di copia per copiare il nome host DNS. Assicurati anche di annotare il numero di porte.
3. Avvia un emulatore di terminale. Questo tutorial utilizza Micro Focus Rumba+.

   > **Note**
   >
   > I passaggi di configurazione variano a seconda dell'emulatore.

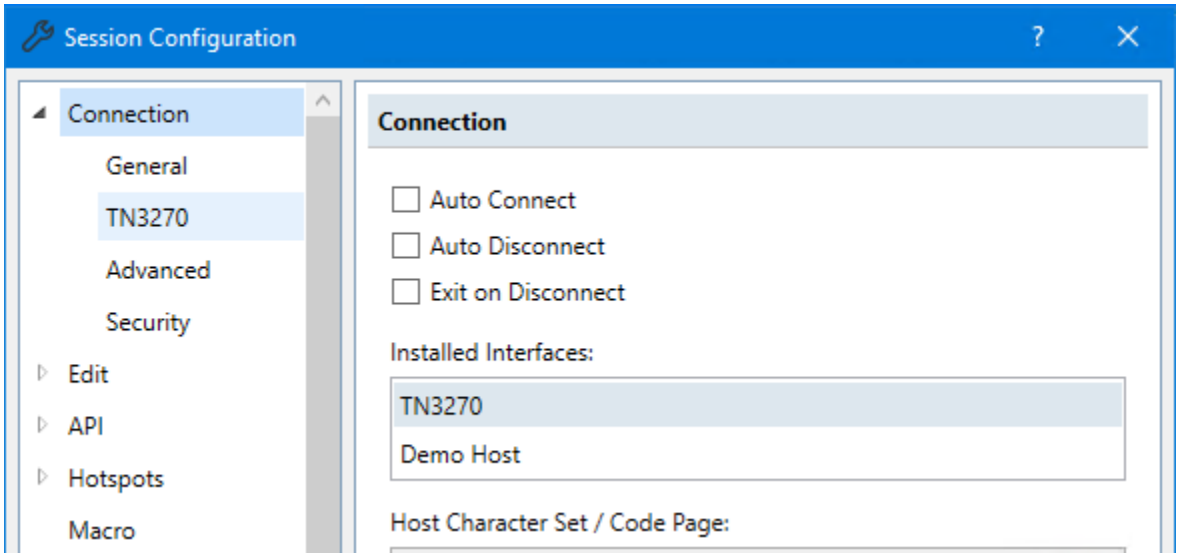
4. Scegli Mainframe Display.



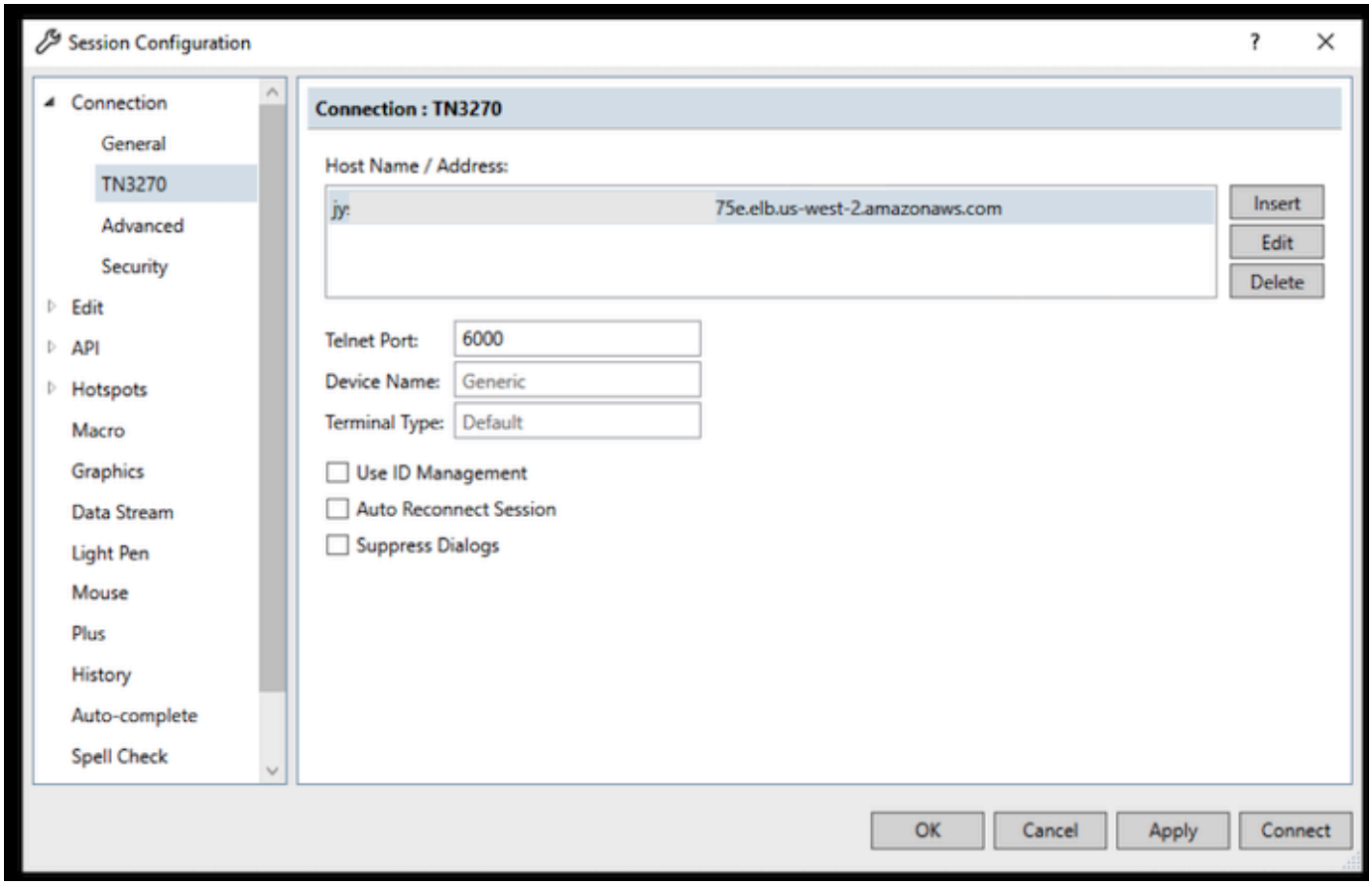
5. Scegli Connessione, quindi scegli Configura.



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



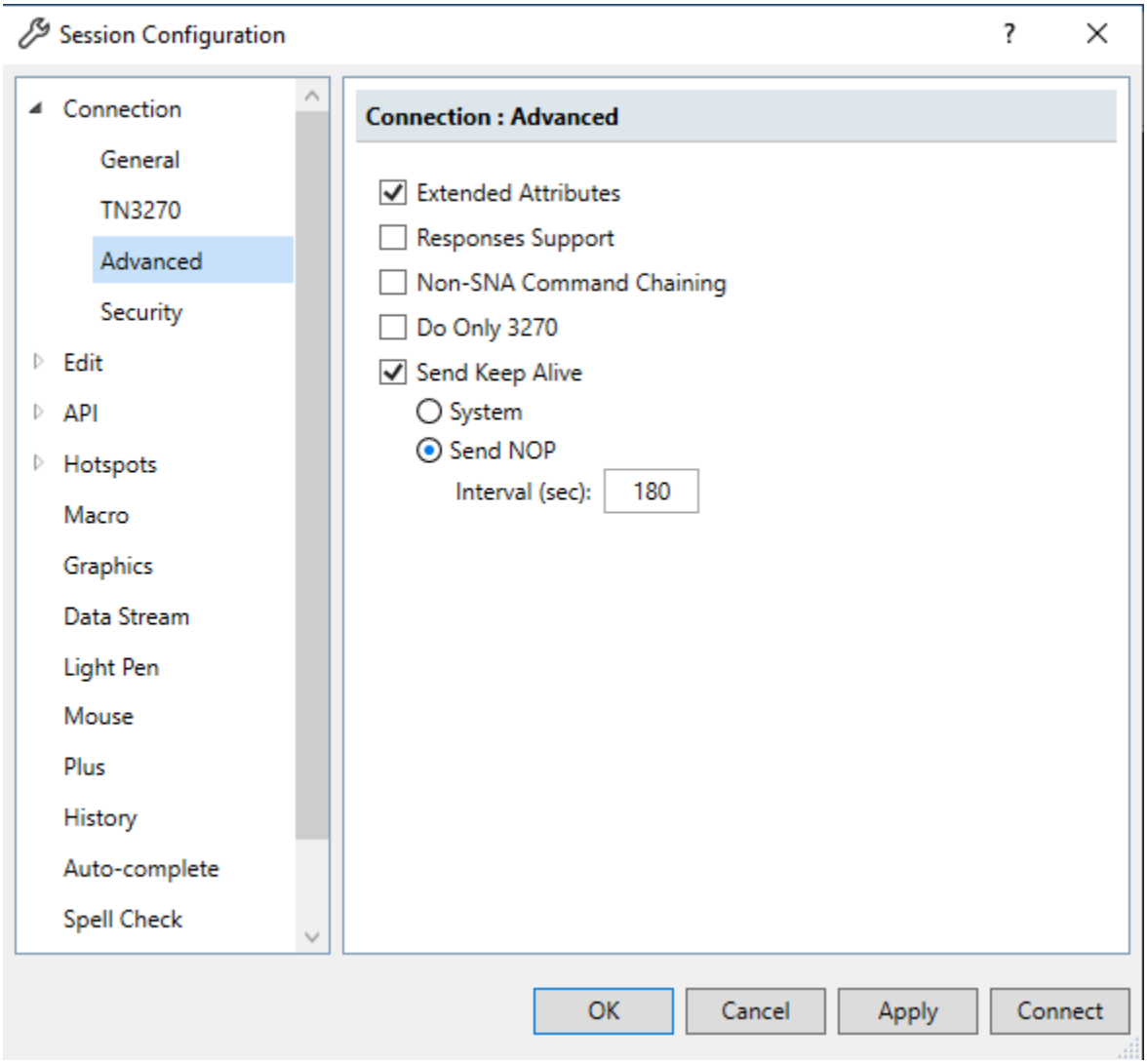
7. Scegliete Inserisci e incollate il file DNS `Hostname` per l'applicazione. `6000`Specificate la porta Telnet.



> **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 TN3270 terminale su almeno 180 secondi aiuta a garantire che Network Load Balancer non interrompa la connessione.



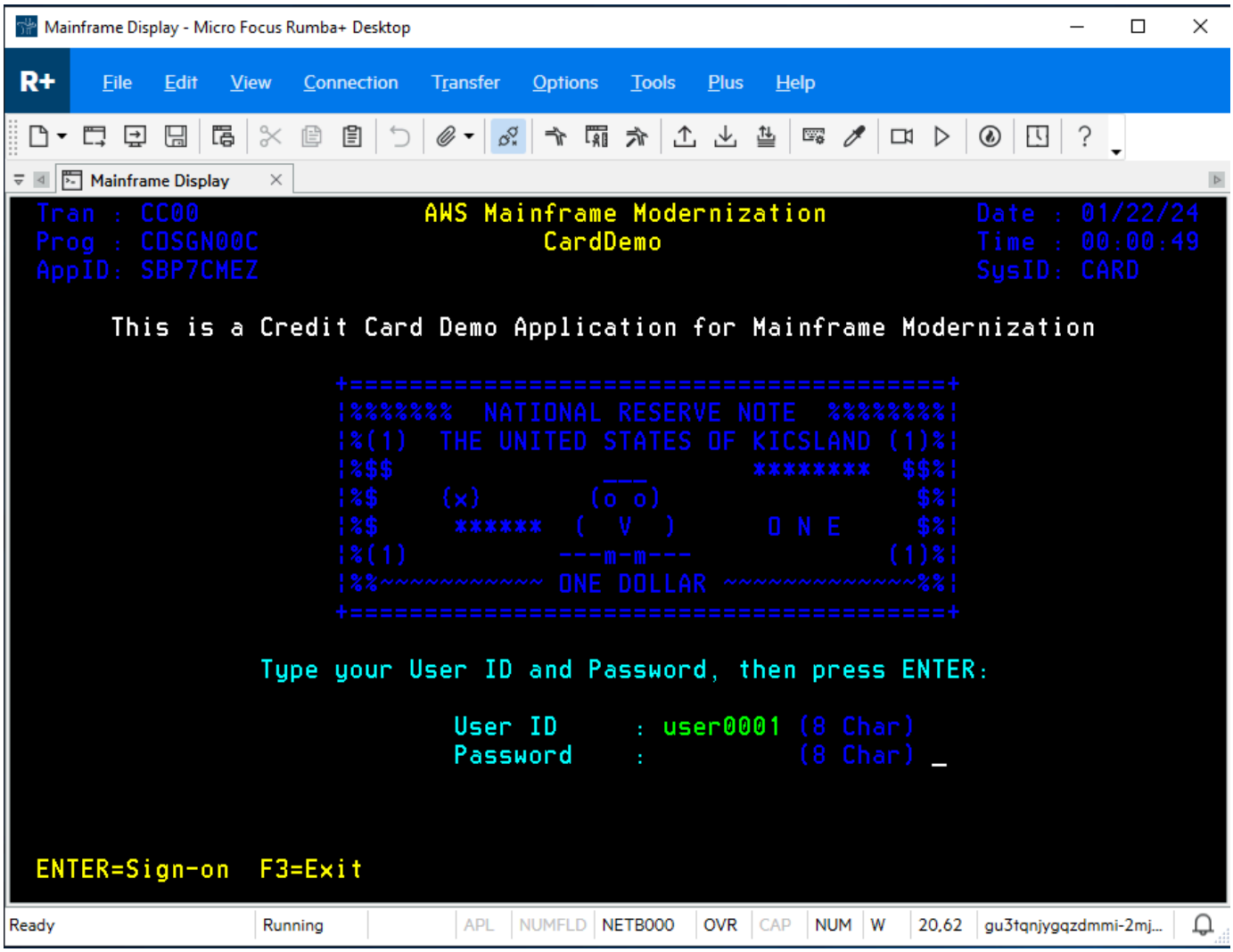
9. Scegli Connetti.

> **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

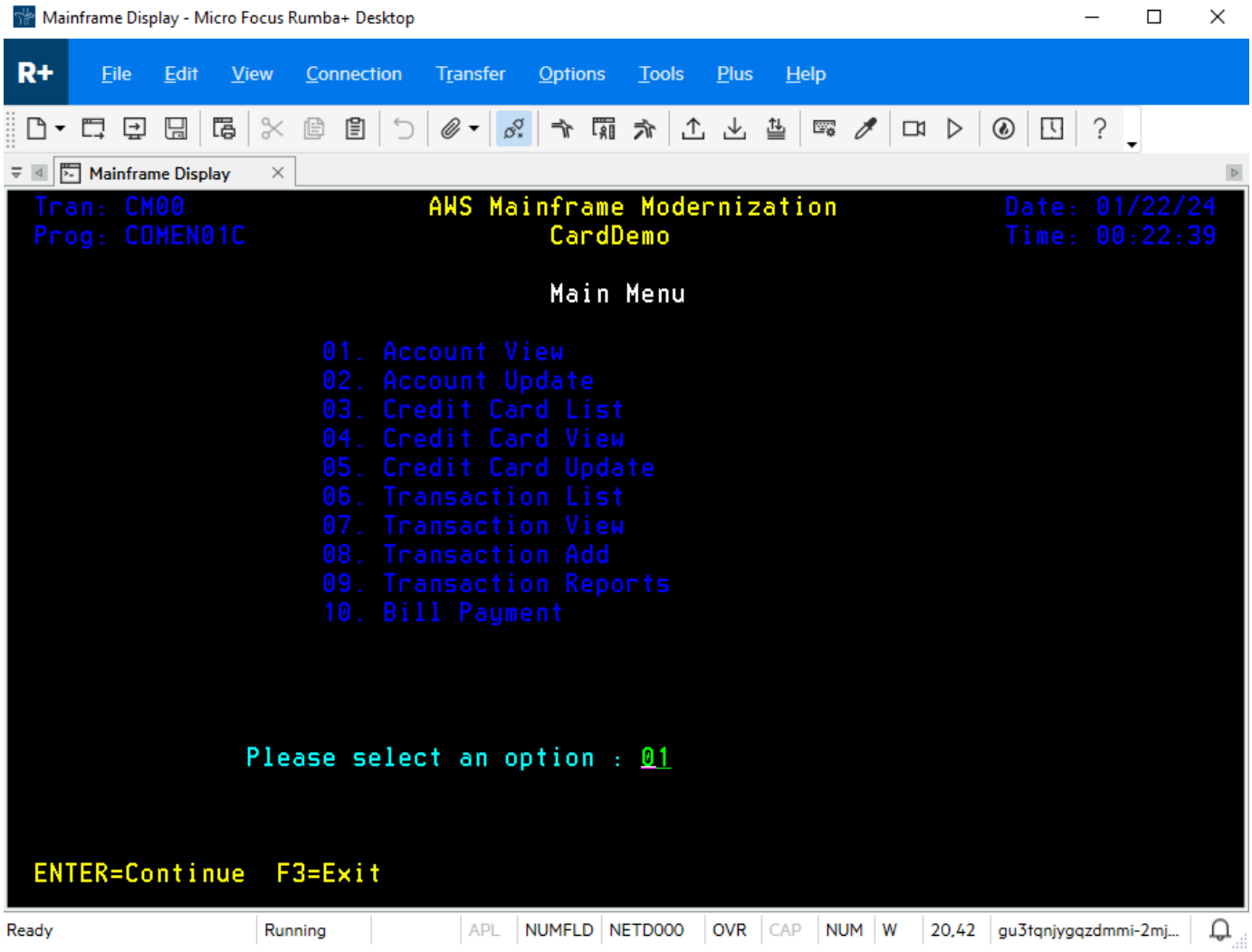
porta 6000 da qualsiasi luogo (ad esempio CIDR Block 0.0.0.0/0). Se ti connetti con successo, allora sai che il gruppo di sicurezza stava bloccando il tuo traffico. Cambia l'origine del gruppo di sicurezza con qualcosa di più specifico. Per ulteriori informazioni sui gruppi di sicurezza, consulta Nozioni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

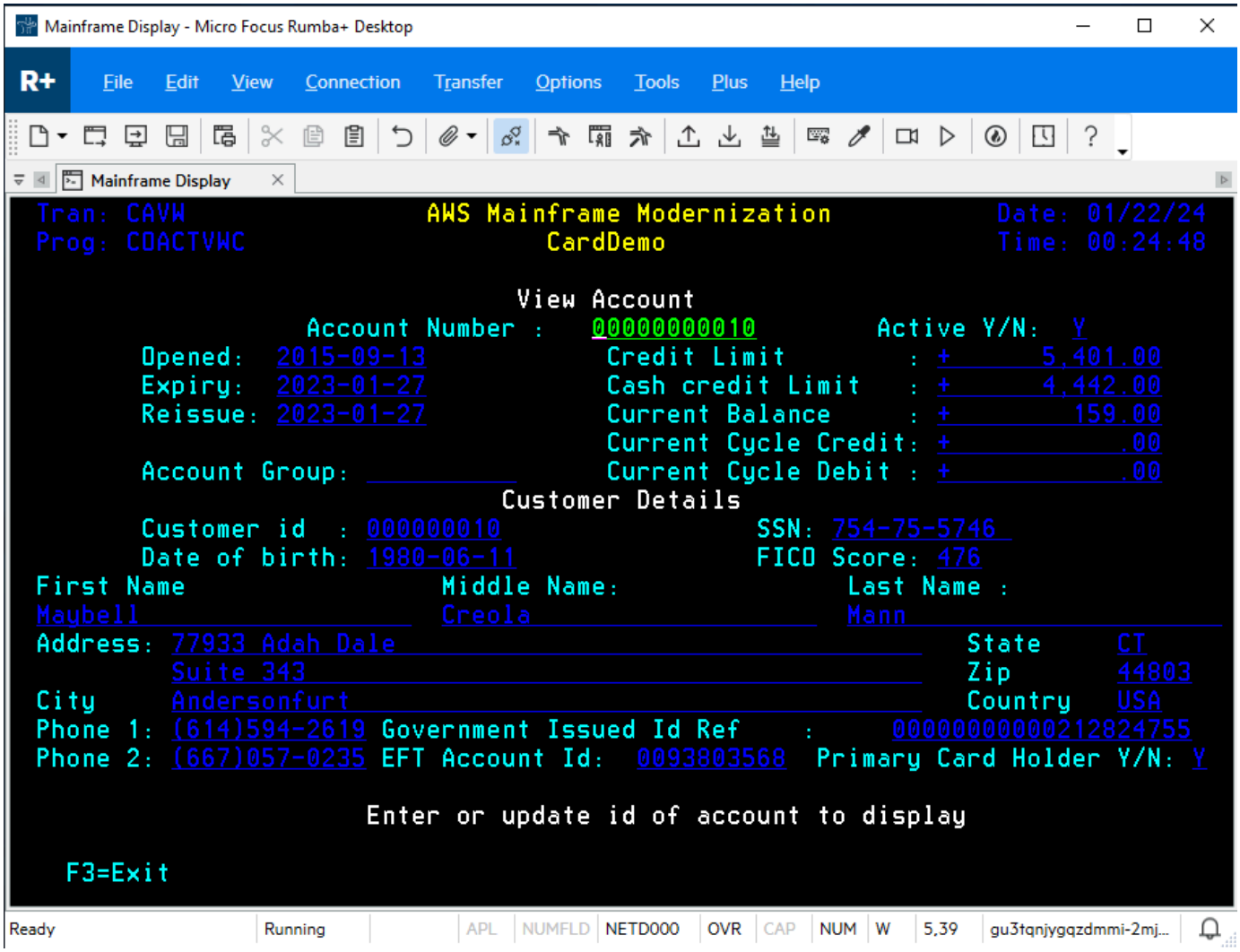


11. Dopo aver effettuato correttamente l'accesso, puoi navigare all'interno dell' CardDemoapplicazione.
12. Accedi `01` alla visualizzazione dell'account.



13. Inserisci `00000000010` il numero di conto e premi Invio sulla tastiera.

> **Note**
>
> Altri account validi sono `0000000011` e`00000000020`.



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

Eseguire la 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 Modernizzazione del mainframe AWS.
- 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 Gestione dei segreti AWS 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.

Fasi successive

Per informazioni su come configurare un ambiente di sviluppo per le applicazioni modernizzate, consulta il Tutorial: Configurazione AppStream 2.0 per l'uso con Rocket Enterprise Analyzer e Rocket Enterprise Developer.

AWS Ciclo di vita dei componenti di modernizzazione del mainframe

Ogni componente di AWS Mainframe Modernization viene sottoposto ad aggiornamenti di versione e a un ciclo di vita di sviluppo. È possibile utilizzare questa pagina come panoramica per comprendere questi componenti, i relativi piani di aggiornamento delle versioni e il modo in cui AWS Mainframe Modernization comunica il rilascio o l'obsolescenza di questi componenti o delle relative versioni.

Panoramica del ciclo di vita dei componenti

AWS Il ciclo di vita della modernizzazione del mainframe descrive l'approccio e le tempistiche per il rilascio e il supporto dei componenti del servizio di modernizzazione del mainframe durante il loro ciclo di vita. AWS Fornire un ciclo di vita prevedibile e coerente vi aiuta a pianificare, testare e implementare le versioni più recenti.

Tutti i componenti AWS di modernizzazione del AWS mainframe forniti traggono vantaggio dal supporto del prodotto fornito Supporto dal momento del rilascio fino al loro ritiro, secondo la tabella del calendario di rilascio di ciascun componente. Per saperne di più sull' Supporto ambito e sulle attività, consulta Compare Plans. Supporto Durante i progetti di modernizzazione attiva, in genere incoraggiamo che l'assistenza clienti venga fornita innanzitutto da team di fornitura di servizi professionali, come indicato nella dichiarazione di lavoro.

AWS Mainframe Modernization rilascia alcuni componenti con versioni provenienti da fornitori che possono essere essi AWS stessi, AWS partner selezionati o comunità. Per ogni componente di modernizzazione AWS del mainframe, una versione ha un numero di versione principale e un numero di versione secondario. Ogni componente ha la propria numerazione delle versioni principali e secondarie.

Per i componenti con versione, abbiamo i seguenti intenti:

- Rilasciare le versioni più recenti dei componenti di modernizzazione del AWS mainframe regolarmente o in base alla richiesta del cliente. Se si desidera una versione più recente di un componente e non è ancora disponibile nel servizio di modernizzazione del AWS mainframe, è possibile effettuare una richiesta esplicita tramite Supporto Product Feature Request (PFR).
- Per allineare le date di fine del supporto e di ritiro delle versioni specifiche del componente AWS Mainframe Modernization con le date di fine del supporto del fornitore del componente.
- Informare i clienti circa un anno prima del ritiro della versione principale di un componente.

Sebbene ci impegniamo a rispettare queste linee guida, in alcuni casi potremmo ritirare versioni specifiche prima con tempi di notifica più brevi. Ad esempio, potremmo ritirare tempestivamente una versione con problemi di sicurezza con un periodo di notifica più breve. Potremmo anche ritirare anticipatamente le versioni secondarie quando una versione secondaria presenta bug o problemi di sicurezza significativi che sono stati risolti in una versione secondaria successiva. Nell'improbabile eventualità che si verifichino tali casi, informeremo i clienti e comunicheremo il piano e la tempistica per il pensionamento. Circostanze specifiche possono imporre scadenze diverse a seconda della situazione.

> **Note**
>
> Gli aggiornamenti critici dei componenti potrebbero essere resi disponibili in qualsiasi momento. Ad esempio, le nuove versioni possono essere rese disponibili tempestivamente per motivi di sicurezza o per fornire correzioni agli ambienti di produzione. Per le richieste effettuate Supporto, il piano di supporto stabilisce i processi, la gravità e i tempi di risposta.

Quando una versione di un componente viene ritirata, AWS Mainframe Modernization non distribuisce queste versioni ai clienti per nuove implementazioni. Di conseguenza, anche queste versioni non sono supportate da. Supporto I clienti che eseguono implementazioni di componenti esistenti dopo la data di ritiro delle versioni devono essere consapevoli dei rischi derivanti da tale operazione. AWS non è responsabile della fornitura di aggiornamenti di sicurezza, supporto tecnico o correzioni rapide per le versioni dei componenti ritirate. Inoltre, non rimuoviamo automaticamente l'accesso o eliminiamo le risorse dell'ambiente. Ti consigliamo vivamente di verificare la disponibilità di nuove versioni ogni 3 mesi e di aggiornare tutti i componenti di modernizzazione del AWS mainframe alle versioni più recenti supportate.

Aggiornamento della versione

AWS Transform for Mainframe Refactor rilascia regolarmente nuove versioni del suo motore di trasformazione e dei componenti di supporto. Questi aggiornamenti forniscono correzioni di bug, miglioramenti della sicurezza, miglioramenti delle prestazioni, supporto esteso della piattaforma e nuove funzionalità di modernizzazione. Consigliamo vivamente di eseguire l'aggiornamento regolarmente per sfruttare le correzioni, le funzionalità e gli aggiornamenti di sicurezza più recenti. Quando una nuova versione diventa disponibile, puoi decidere se e quando applicarla ai tuoi progetti di trasformazione e agli artefatti generati. Le versioni rilasciate seguono uno schema di numerazione Major.Minor.Patch e rientrano in tre categorie:

- Le versioni principali introducono nuove funzionalità o modifiche strutturali significative. Queste includono in genere:
  - Support per piattaforme legacy aggiuntive (ad esempio Fujitsu GS21).
  - Supporto per nuovi linguaggi legacy (ad esempio Natural).
  - Nuovo supporto per database legacy (ad esempio, Adabas, Fujitsu NDB).
  - Principali aggiornamenti delle versioni dello stack di modernizzazione di destinazione (ad esempio, versioni più recenti di Spring, Java o PostgreSQL).

  Le versioni principali possono introdurre modifiche sostanziali che richiedono aggiustamenti alle applicazioni modernizzate esistenti, in particolare quando si rigenera il codice o si adatta alle definizioni aggiornate dell'architettura di destinazione.
- Le versioni minori offrono miglioramenti evolutivi, correzioni di bug e aggiornamenti di manutenzione. Mantengono invariati sia le platform/language/database definizioni legacy supportate che lo stack di modernizzazione target.

  Tuttavia, le versioni minori possono ancora includere modifiche sostanziali nel codice sorgente generato o nelle classi di runtime per scopi di utilizzo interno. Ciò può influire sulle applicazioni già modernizzate che sono state sottoposte a manutenzione manuale post-generazione e che non possono essere rigenerate in sicurezza dalle fonti legacy originali senza ulteriori rielaborazioni.
- Le versioni delle patch fungono da canale di emergenza per risolvere rapidamente le regressioni critiche o le correzioni urgenti introdotte nelle versioni recenti, preservando al contempo la compatibilità. Hanno un ambito ristretto e mirano a ripristinare la stabilità con il minimo rischio.

Per ridurre al minimo i rischi durante qualsiasi aggiornamento:

- Esegui sempre test completi non di regressione prima di applicare modifiche di versione a progetti live o legati alla produzione.
- Sfrutta le pipeline DevOps di test e implementazione (idealmente create durante l'iniziativa di modernizzazione) per automatizzare la convalida del codice generato e del comportamento di runtime dopo gli aggiornamenti.
- Prendi in considerazione le strategie blue/green di implementazione per gli ambienti di runtime quando aggiorni i componenti.

Per ulteriori indicazioni sulle strategie di implementazione e sulla gestione delle modifiche, consulta AWS Well-Architected Reliability Pillar.

AWS Panoramica della versione di Mainframe Modernization Refactor with Transform for mainframe AWS

Con AWS Transform for mainframe runtime, la versione segue uno schema. `Major.Minor.Patch` Ad esempio, per la versione runtime di AWS Transform for mainframe`4.1.0`, la versione principale è 4, la versione secondaria è 1 e la versione patch è 0.

Intendiamo rilasciare nuove versioni principali di AWS Transform for mainframe runtime in caso di modifiche importanti al runtime o alle relative dipendenze. AWS Le versioni principali di Transform for mainframe runtime sono supportate per almeno 12 mesi, a meno che non compaiano alcune vulnerabilità ed esposizioni comuni (). CVEs Il supporto copre i bug nelle funzionalità di runtime, come indicato nella nostra documentazione. Nel caso di Critical e High CVEs nelle dipendenze del runtime (Spring, Java, Tomcat e altre), la durata del supporto della versione principale è ridotta a 6 mesi per High CVEs e a 3 mesi per Critical CVEs dalla data di rilascio della nuova versione di runtime che corregge il CVE, salvo diversa indicazione esplicita.

Intendiamo rilasciare mensilmente nuove versioni secondarie di AWS Transform for mainframe. I clienti sono tenuti ad aggiornare regolarmente le versioni per ottenere le ultime correzioni di sicurezza, correzioni di bug e miglioramenti delle funzionalità. I progetti attivi non ancora in produzione devono adottare l'ultima versione di runtime non appena sarà disponibile.

Le nuove correzioni vengono fornite nell'ultima versione secondaria per la particolare versione principale in cui viene sollevato un problema. Se sono necessarie nuove correzioni, è necessario eseguire l'aggiornamento a una nuova versione secondaria per applicarle.

Le versioni con patch per le versioni supportate vengono fornite solo per risolvere difetti di runtime critici non presenti nelle precedenti versioni secondarie supportate.

Le pre-release Alpha sono versioni di breve durata rese disponibili per un'iterazione rapida durante la consegna dei progetti. Le correzioni per i problemi identificati nelle versioni preliminari alfa vengono fornite nelle versioni secondarie successive, poiché non viene fornita alcuna patch per le versioni Alpha pre-release.

È possibile trovare le date di rilascio e i dettagli su ciascuna versione di runtime nel. the section called "AWS Note sulla versione di Transform for mainframe"

Le scansioni di sicurezza vengono eseguite da Amazon Inspector.

Comprendi le applicazioni gestite in Modernizzazione del mainframe AWS

Se non conosci Modernizzazione del mainframe AWS i seguenti argomenti per iniziare:

- Cos'è la modernizzazione del AWS mainframe?
- Configurazione per la modernizzazione del AWS mainframe
- Tutorial: configurare il runtime gestito per AWS Transform for mainframe
- Tutorial: Configurazione del runtime gestito per Rocket Software (precedentemente Micro Focus)

Un'applicazione Modernizzazione del mainframe AWS 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 Modernizzazione del mainframe AWS applicazioni per ogni carico di lavoro da migrare. Quando si crea un' Modernizzazione del mainframe AWS applicazione, si specifica il motore su cui viene eseguita l'applicazione al momento della creazione. Scegliete AWS Transform for mainframe se utilizzate il pattern di refactoring automatico e scegliete Rocket Software (precedentemente Micro Focus) se utilizzate il pattern di replatforming.

Argomenti

- Crea risorse per un'applicazione migrata AWS
- Crea un'applicazione Modernizzazione del mainframe AWS
- Distribuisci un' Modernizzazione del mainframe AWS applicazione
- Aggiornare un'applicazione Modernizzazione del mainframe AWS
- Eliminare un'applicazione Modernizzazione del mainframe AWS
- Invia lavori in batch per Modernizzazione del mainframe AWS le candidature
- Annulla i processi in batch per le applicazioni Modernizzazione del mainframe AWS
- Importa set di dati per Modernizzazione del mainframe AWS le applicazioni
- Esporta set di dati per le applicazioni Modernizzazione del mainframe AWS
- Gestisci le transazioni per le applicazioni Modernizzazione del mainframe AWS
- Configurare l'applicazione gestita da Rocket Software (precedentemente Micro Focus)
- Configurare l' AWS applicazione gestita Transform for mainframe

- Modernizzazione del mainframe AWS riferimento alla definizione dell'applicazione
- AWS Riferimento alla definizione del set di dati per la modernizzazione del mainframe

Crea risorse per un'applicazione migrata AWS

Per eseguire l'applicazione migrata 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 Gestione dei segreti AWS 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 rifattorizzate che quelle riplatformate richiedono un bucket Amazon 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 nella definizione dell'applicazione, creata come parte dell'applicazione. Modernizzazione del mainframe 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. Modernizzazione del mainframe AWS 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 Gestione dei segreti AWS segreto.

Se utilizzi il pattern di refactoring AWS Transform for mainframe e hai bisogno di un Blusam database, il motore di runtime AWS Transform for mainframe 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 pattern di replatforming di Rocket Software, 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 Gestione dei segreti AWS utilizzando un per crittografarle. AWS KMS key

AWS Key Management Service chiave

È necessario archiviare le credenziali per il database dell'applicazione in modo sicuro in. Gestione dei segreti AWS Per creare un segreto in Secrets Manager, devi creare un AWS 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, devi aggiornare la politica della chiave per concedere le autorizzazioni di Modernizzazione del mainframe AWS decrittografia. Aggiungi le seguenti dichiarazioni politiche:

```
{
   "Effect" : "Allow",
   "Principal" : {
   "Service" : "m2.amazonaws.com"
   },
   "Action" : "kms:Decrypt",
   "Resource" : "*"
   }
```

Gestione dei segreti AWS segreto

È necessario archiviare le credenziali per il database dell'applicazione in modo sicuro in. Gestione dei segreti AWS Per creare un segreto, seguite la procedura descritta in Creare un segreto del database nella Guida per l'Gestione dei segreti AWS utente.

Modernizzazione del mainframe AWS 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-full`, `verify-ca`, o`disable`.

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" : "*"
    }
```

Crea un'applicazione Modernizzazione del mainframe AWS

Usa la Modernizzazione del mainframe AWS console per creare un' Modernizzazione del mainframe AWS applicazione. La creazione di un'applicazione consente di eseguire attività con il carico di lavoro del mainframe migrato.

Queste istruzioni presuppongono che la procedura in Configurazione per la modernizzazione del AWS mainframe sia stata completata.

Creazione di un'applicazione

Come creare un'applicazione

1. Apri la console all' Modernizzazione del mainframe AWS indirizzo. https://console.aws.amazon.com/m2/
2. Nel Regione AWS selettore, scegli la regione in cui desideri 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, scegli AWS Transform per mainframe per il refactoring automatizzato o Micro Focus (Rocket) per il riplatforming.
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 modernizzazione AWS del mainframe.

> **Note**
>
> Per impostazione predefinita, Modernizzazione del mainframe AWS crittografa i tuoi dati con una AWS KMS chiave che Modernizzazione del mainframe AWS possiede e

gestisce per te. Tuttavia, puoi scegliere di utilizzare una AWS KMS chiave gestita dal cliente.

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. Scegli Next (Successivo).
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 Transform for mainframe o Definizione dell'applicazione Rocket Software (precedentemente Micro Focus).
12. Scegli Next (Successivo).
13. Nella pagina Rivedi e crea, esamina le informazioni che hai inserito, quindi scegli Crea applicazione.

Distribuisci un' Modernizzazione del mainframe AWS applicazione

Usa la Modernizzazione del mainframe AWS console per distribuire un'applicazione. Modernizzazione del mainframe AWS È necessario distribuire le applicazioni in un ambiente di runtime prima di eseguire le attività.

Queste istruzioni presuppongono che la procedura in Configurazione per la modernizzazione del AWS mainframe sia stata completata.

Distribuisci un'applicazione

Per eseguire un' Modernizzazione del mainframe AWS applicazione, 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 distribuisci 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 Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui desideri 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 Implementa.

Per distribuire una versione diversa di un'applicazione distribuita

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo. https://console.aws.amazon.com/m2/
2. Nel Regione AWS selettore, scegli la regione in cui desideri 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 Implementa.

Aggiornare un'applicazione Modernizzazione del mainframe AWS

Usa la Modernizzazione del mainframe AWS console per aggiornare un' Modernizzazione del mainframe AWS applicazione. L'aggiornamento di un'applicazione crea una nuova versione dell'applicazione.

Queste istruzioni presuppongono che la procedura in Configurazione per la modernizzazione del AWS mainframe sia stata completata.

Aggiornamento di un'applicazione

Un' Modernizzazione del mainframe AWS applicazione 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.

Come aggiornare un'applicazione

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione che desideri 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 Transform for mainframe o Definizione dell'applicazione Rocket Software (precedentemente 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 Distribuisci un' Modernizzazione del mainframe AWS applicazione.

Eliminare un'applicazione Modernizzazione del mainframe AWS

Puoi eliminare un' Modernizzazione del mainframe AWS applicazione da un ambiente utilizzando la Modernizzazione del mainframe AWS console.

Queste istruzioni presuppongono che la procedura in Configurazione per la modernizzazione del AWS mainframe sia stata completata.

Eliminazione di un'applicazione

Se devi eliminare un' Modernizzazione del mainframe AWS applicazione che è in esecuzione, assicurati di interromperla prima. Puoi vedere lo stato dell'applicazione nella pagina Applicazioni.

Come eliminare un'applicazione

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione che desideri 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.

Invia lavori in batch per Modernizzazione del mainframe AWS le candidature

In Modernizzazione del mainframe AWS 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, Modernizzazione del mainframe AWS viene creata un'esecuzione del processo 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, utilizza la Modernizzazione del mainframe AWS console per importare i set di dati. Per ulteriori informazioni, consulta Importa set di dati per Modernizzazione del mainframe AWS le applicazioni.

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali Configurazione per la modernizzazione del AWS mainframe e successiviCrea un'applicazione Modernizzazione del mainframe AWS.

Argomenti

- Invia un lavoro in batch
- Riavviare un processo batch

Invia un lavoro in batch

Per inviare un lavoro in batch

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione per la quale desideri 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).

Riavviare un processo batch

Per riavviare un processo batch

> **Important**
>
> Il riavvio di un processo in batch è disponibile nelle seguenti versioni del motore:
>
> - Motore ambientale Micro Focus (Rocket) versioni 8.0.6 o successive. È inoltre necessario disporre di un EFS o di un FSx file system collegato all'ambiente.
> - AWS Versioni del motore Transform for mainframe Environment 4.3.0 o successive. È inoltre necessario disporre di un EFS o di un FSx file system collegato se si tratta di un ambiente HA.

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui sono stati creati l'applicazione e il processo batch.
3. Nella pagina Applicazioni, scegli l'applicazione in cui desideri riavviare un processo batch.
4. Nella pagina dei dettagli dell'applicazione, scegli Lavori in Batch.
5. Seleziona il processo batch che desideri riavviare dall'elenco generato. Vai al menu Azioni e scegli Riavvia processo.
6. Specificate come desiderate riavviare il processo batch. È possibile effettuare le seguenti operazioni per il motore di ambiente Micro Focus (Rocket) e il motore di ambiente AWS Transform for mainframe:

   - Per il motore ambientale Micro Focus (Rocket), è possibile scegliere di riavviare dall'inizio o riavviare utilizzando step o procsteps.
     - L'opzione Riavvia dall'inizio consente di riavviare tutti i passaggi di un processo batch dall'inizio.
     - L'opzione Riavvia utilizzando steps o procsteps consente di scegliere un passaggio o un passaggio di procedura specifico da riavviare e, facoltativamente, un passaggio o un passaggio della procedura che si desidera terminare.

> **Note**
>
> Il passaggio finale o procstep deve essere maggiore o uguale al numero del passo iniziale o del procstep.

- Per il motore di ambiente AWS Transform for mainframe, è possibile riavviare l'esecuzione più recente di un processo batch da un JCL/PROC passaggio precedentemente non riuscito oppure eseguire un riavvio ritardato ignorando i passaggi precedentemente eseguiti con successo.
  - È possibile scegliere un nome specifico per il passaggio che si desidera riavviare.
  - Facoltativamente, puoi utilizzare Skip step per ignorare il passaggio selezionato e riavviare dal passaggio successivo del flusso di lavoro.

7. Seleziona Submit job (Invia processo).

Annulla i processi in batch per le applicazioni Modernizzazione del mainframe AWS

In Modernizzazione del mainframe AWS puoi annullare i lavori in batch per le tue applicazioni. È possibile esaminare i dettagli sulle esecuzioni dei processi in batch. Ogni volta che si invia un processo batch, Modernizzazione del mainframe AWS viene creata un'esecuzione del processo 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.

Annullamento di un processo in batch

Quando si annulla un processo batch, non viene eliminato un processo batch, ma l'esecuzione delle attività relative a quel processo batch. È comunque possibile visualizzare i dettagli del processo batch.

Per annullare un processo in batch

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione con l'applicazione per i lavori in batch.
3. Dall'elenco dei processi batch, trova e seleziona il processo batch che desideri annullare.
4. Scegli Azioni e scegli Annulla lavoro.
5. Scegli Annulla processo batch.

In questo modo verranno annullate tutte le attività di lavoro in batch che avevi programmato per l'esecuzione.

Importa set di dati per Modernizzazione del mainframe AWS le applicazioni

Con Modernizzazione del mainframe AWS puoi importare set di dati da utilizzare con le tue applicazioni. Puoi specificare i set di dati da importare 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 Modernizzazione del mainframe AWS console per importare set di dati per un' Modernizzazione del mainframe AWS applicazione.

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali Configurazione per la modernizzazione del AWS mainframe e successiviCrea un'applicazione Modernizzazione del mainframe AWS.

Importa un set di dati

Per importare un set di dati

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione per cui desideri 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. Scegli Importa.
6. Esegui una delle seguenti 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 "Data set definition reference" 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.

Esporta set di dati per le applicazioni Modernizzazione del mainframe AWS

Con Modernizzazione del mainframe AWS puoi esportare set di dati da utilizzare con le tue applicazioni. Puoi specificare i set di dati da esportare in un file JSON archiviato in un bucket Amazon S3 oppure puoi specificare i valori di configurazione dei set di dati separatamente. Dopo aver esportato i set di dati, puoi esaminare i dettagli dell'attività di esportazione per confermare che i set di dati che desideri siano stati esportati.

Utilizzate la Modernizzazione del mainframe AWS console per esportare i set di dati per un' Modernizzazione del mainframe AWS applicazione.

Queste istruzioni presuppongono che tu abbia completato i passaggi iniziali Configurazione per la modernizzazione del AWS mainframe e successiviCrea un'applicazione Modernizzazione del mainframe AWS.

Esportazione di un set di dati

Per esportare un set di dati

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione per cui desideri importare i set di dati.
3. Nella pagina Applicazioni, scegli l'applicazione per cui desideri esportare i set di dati.
4. Nella pagina dei dettagli dell'applicazione, scegli Set di dati.
5. Scegli Export (Esporta).
6. Esegui una delle seguenti 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. Per ulteriori informazioni, consulta the section called "Data set definition reference".

     Inserisci il nome del set di dati, 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 Modernizzazione del mainframe AWS

Con Modernizzazione del mainframe AWS 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 per la modernizzazione del AWS mainframe e successiviCrea un'applicazione Modernizzazione del mainframe AWS.

Gestisci le transazioni per le applicazioni

Per gestire le transazioni per le applicazioni

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nel Regione AWS selettore, scegli la regione in cui è stata creata l'applicazione che desideri eseguire.
3. Nella pagina Applicazioni, scegli l'applicazione in 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. Puoi visualizzare le risorse in base alle risorse delle transazioni, ai gruppi, agli elenchi o SITs.

   - 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**
     >
     > Il Modernizzazione del mainframe AWS servizio supporta tipi di risorse aggiuntivi per gestire le transazioni per le applicazioni ed è accessibile dalla console.

   - I gruppi sono una raccolta di risorse per le transazioni. Puoi scegliere i gruppi che desideri 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 AWS Transform for mainframe refactor engine, si specificano gli elenchi da includere all'avvio. Non c'è limite al numero di elenchi.
     - Con il motore di ripiattaforma Rocket Software, puoi specificare fino a quattro elenchi in un SIT.

- SIT (System Initialization Table) mostra tutte le configurazioni di transazione disponibili. È possibile trovarle SITs in base alle proprietà (nome, descrizione ed elenchi di avvio). Puoi anche scegliere elenchi da associare al SIT prescelto.

> **Note**
>
> SITs sono applicabili solo al motore di ripiattaforma Rocket Software.

5. Scegli una risorsa di transazione per visualizzare tutte le informazioni sulla risorsa. È inoltre possibile visualizzare tutti gli attributi associati alla risorsa di transazione.

Configurare l'applicazione gestita da Rocket Software (precedentemente Micro Focus)

Puoi configurare le tue applicazioni con il motore di runtime Rocket Software per personalizzare proprietà aggiuntive, incluse le integrazioni.

Argomenti

- Integrazioni di terze parti supportate per Rocket Software
  - Stampanti

Integrazioni di terze parti supportate per Rocket Software

Per utilizzare integrazioni di terze parti, l'ambiente gestito di modernizzazione del AWS mainframe deve utilizzare una versione del motore Rocket Software che supporti questo tipo di configurazione. Sono supportate le versioni del motore con il suffisso R (ad esempio, la versione 9.0.9.R). Ciò significa che la versione 9.0.9.R del motore include il supporto per l'installazione del client per le integrazioni di parti, ma la 9.0.9 no.

Stampanti

Le risorse della stampante sono configurate tramite la definizione dell'applicazione Rocket Software, come descritto nella sezione. the section called “Stampanti: opzionali”

Una definizione di stampante può definire un modulo di uscita personalizzato o fornito dal servizio per la stampante. Alcuni esempi di possibili configurazioni del modulo di uscita sono:

1. Esempio di servizio di caricamento binario fornito.

```
...
{
    "name": "p1",
    "classes": [
        "AB"
    ],
    "description": "Using service managed LRS Queue exit module",
    "exit-module": {
        "name": "lrsprte6"
    }
},
...
```

2. Esempio di fornitura di file binari da S3.

```
"exit-module": {
        "name": "s3Exit",
        "module": "${s3-source}/3pa/s3Exit.so"
}
```

3. Esempio di fornitura di file binari da EFS.

> **Note**
>
> Per utilizzare EFS mount, è necessario collegarlo durante la creazione dell'ambiente, insieme ad alcuni valori aggiuntivi da impostare come`program-path`.

```
...
"batch-settings": {
    "jes-printers": [
        {
            "name": "p3",
            "classes": [
                "EF"
            ],
            "description": "Using binary from customer provided exit module on EFS
 Mount",
            "exit-module": {
```

```
                "name": "efsExit"
            }
        }
    ],
    "program-path": "$EFS_MOUNT/path/to/directory/containing/binaries/"
},
"runtime-settings": {
    "environment-variables": {
        "EFS_MOUNT": "/m2/mount/efs"
    }
}
...
```

Coda LRS: opzionale

Per utilizzare la coda LRS, è necessario utilizzare un motore Rocket Software che supporti artefatti di terze parti (ad esempio motori che terminano con). `.R` Oltre a configurare una stampante con un modulo di uscita che punti `lrsprte6` come nome di immissione del modulo di uscita, la coda LRS richiede una variabile di ambiente aggiuntiva, come definita nel blocco preesistente «runtime-settings» della definizione dell'applicazione Rocket Software.

LRSQ_ADDRESS

(Obbligatorio) Specificate l'indirizzo del server LRS a cui inviare il modulo di uscita di stampa LRSQ.

Stampanti LRS - La configurazione di una stampante LRS richiede la definizione di una stampante jes come specificato nella sezione. the section called "Stampanti: opzionali"

Inoltre, LRSQ_ADDRESS deve essere specificato come parte del campo nella definizione dell'applicazione. `runtime-settings`

```
"runtime-settings": {
    "environment-variables": {
        "LRSQ_ADDRESS": "<lrsq-address>"
    }
}
```

Configurare l' AWS applicazione gestita Transform for mainframe

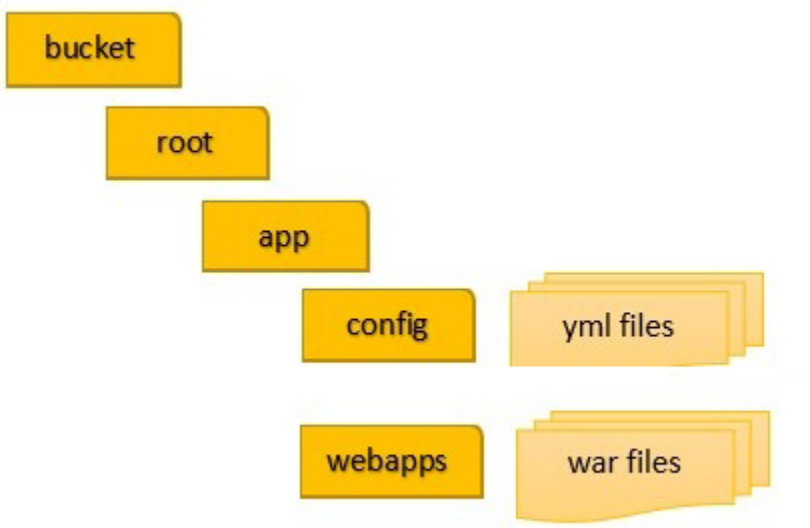
È possibile configurare l'applicazione in modo da includere l'accesso alle utilità legacy. È possibile personalizzare anche proprietà aggiuntive. Per capire cosa è possibile configurare e dove, consulta la the section called "Struttura di AWS Transform per applicazioni gestite da mainframe" sezione per comprendere la struttura generale di un'applicazione modernizzata AWS Transform for mainframe.

Argomenti

- Struttura di AWS Transform per applicazioni gestite da mainframe
- Configurare l'accesso alle utilità per le applicazioni gestite
- Aggiungi proprietà di configurazione per l'applicazione gestita con il motore AWS Transform for mainframe

Struttura di AWS Transform per applicazioni gestite da mainframe

Se utilizzi il pattern di refactoring AWS Transform for mainframe, il motore di runtime AWS Transform for mainframe prevede la seguente struttura nella cartella specificata da nella definizione dell'applicazione: `app-location`



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`

extra

Contiene vasetti che possono fornire funzionalità aggiuntive per l'applicazione modernizzata.

Gestione delle opzioni Java di un'applicazione

Per gestire determinate opzioni Java per l'applicazione, aggiungete un file delle proprietà denominato `tomcat.properties` alla cartella specificata da`app-location`. Questo file può avere tre proprietà:`xms`, che specifica il consumo minimo di memoria Java`xmx`, che specifica il consumo massimo di memoria Java e`dnscachettl`, che gestisce la durata della cache per le risoluzioni DNS. Di seguito è riportato un esempio del contenuto di un file valido. `tomcat.properties`

```
xms=512M
xmx=1G
dnscachettl=5
```

I valori specificati per le prime 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.

Il valore della terza proprietà rappresenta la durata della cache in secondi e può avere il valore -1 (cache per sempre) oppure può variare da 0 (mai memorizzare nella cache) a 999. Nel contesto delle distribuzioni di applicazioni gestite, il valore predefinito è -1.

Configurare l'accesso alle utilità per le applicazioni gestite

Quando si esegue il refactoring di un'applicazione mainframe con AWS Transform for mainframe, 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 essi. AWS Transform for mainframe 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 della 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 della 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) Specificate il set di caratteri utilizzato dal programma di utilità. Nella maggior parte dei casi, quando si esegue la migrazione da z/OS piattaforme, 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) Specificate 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 deve essere una variante EBCDIC. L'impostazione predefinita se non è impostata è. `ASCII`

```
sysPunchEncoding : cp1047
```

Configurazione dell'origine dati

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. Gestione dei segreti AWS Le proprietà che indicano i segreti appropriati in Secrets Manager sono le seguenti:

Fonte di dati primaria

Questo è il database principale delle applicazioni aziendali.

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) Specificate 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
```

spring.aws.client.datasources.primary.type

(Facoltativo) Specifica il nome completo dell'implementazione del pool di connessioni da utilizzare. Il valore predefinito è `com.zaxxer.hikari.HikariDataSource`.

```
spring.aws.client.datasources.primary.type: target-datasource-type
```

Se il tipo di origine dati principale è`com.zaxxer.hikari.HikariDataSource`, è possibile specificare proprietà aggiuntive come segue:

spring.datasource.primary. [nome_proprietà]

(Facoltativo) È possibile utilizzare questo formato per specificare proprietà aggiuntive per la configurazione di un'implementazione del pool di connessioni di origine dati principale.

Di seguito è riportato un esempio di origine dati primaria di tipo`com.zaxxer.hikari.HikariDataSource`.

```
spring:
   datasource:
       primary:
         autoCommit: XXXX
         maximumPoolSize: XXXX
         keepaliveTime: XXXX
         minimumIdle: XXXX
         idleTimeout: XXXX
         connectionTimeout: XXXX
         maxLifetime: XXXX
```

Altre fonti di dati di utilità

Oltre alla fonte di dati principale, è possibile fornire altre fonti di dati di utilità.

spring.aws.client.utility.pgm.datasources.names

(Facoltativo) Specifica l'elenco dei nomi delle fonti di dati di utilità.

```
spring.aws.client.utility.pgm.datasources.names: dsname1, dsname2, dsname3
```

spring.aws.client.utility.pgm.datasources. [dsname] .secret

(Facoltativo) Specifica l'ARN segreto in SSM che ospita le proprietà dell'origine dati. Fornisci [dsname] nell'elenco dei nomi specificato in. `spring.aws.client.utility.pgm.datasources.names`

```
spring.aws.client.utility.pgm.datasources.dsname1.secret: datasource-secret-ARN
```

spring.aws.client.utility.pgm.datasources. [nome ds] .dbname

(Facoltativo) Specificate il nome del database di destinazione se il nome del database non viene fornito direttamente nel segreto del database utilizzando la proprietà. `dbname` Fornire [dsname] nell'elenco dei nomi specificati in. `spring.aws.client.utility.pgm.datasources.names`

```
spring.aws.client.utility.pgm.datasources.dsname1.dbname: target-database-name
```

spring.aws.client.utility.pgm.datasources. [dsname] .type

(Facoltativo) Specifica il nome completo dell'implementazione del pool di connessioni da utilizzare. Il valore predefinito è `com.zaxxer.hikari.HikariDataSource`. Fornire [dsname] nell'elenco dei nomi specificato in. `spring.aws.client.utility.pgm.datasources.names`

```
spring.aws.client.utility.pgm.datasources.dsname1.type: target-datasource-type
```

Se il tipo di origine dati dell'utilità è`com.zaxxer.hikari.HikariDataSource`, è possibile fornire proprietà aggiuntive come segue:

spring.datasource. [nome del file]. [nome_proprietà]

(Facoltativo) Specifica una raccolta di proprietà aggiuntive per configurare l'implementazione di un pool di connessioni con sorgenti di dati di utilità. Fornire [dsname] nell'elenco dei nomi specificati in. `spring.aws.client.utility.pgm.datasources.names` Specificate le proprietà nel formato seguente: `property_name : value`

Di seguito è riportato un esempio di fonti di dati di utilità aggiuntive di tipo`com.zaxxer.hikari.HikariDataSource`:

```
spring:
   datasource:
       dsname1:
         connectionTimeout: XXXX
         maxLifetime: XXXX
       dsname2:
         connectionTimeout: XXXX
         maxLifetime: XXXX
       dsname3:
         connectionTimeout: XXXX
         maxLifetime: XXXX
```

treatLargeNumberAsInteger

(Facoltativo) Relativo alle specifiche del motore di database Oracle e all'utilizzo delle DSNTEP2 DSNTEP4 /utilities. 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 (`+0`to`+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 risorse legacy modernizzate che provengono dalle 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 facoltative.

Questo esempio mostra tutte le possibili proprietà di scaricamento.

```
# 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) Specificate 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) Specificate le proprietà di formattazione di data e ora. Viene utilizzato per gestire gli date/time oggetti nelle query UNLOAD. Il valore predefinito è "`false`".

- Se impostato su`true`, utilizza le `pgmTimestampFormat` proprietà `pgmDateFormatpgmTimeFormat`, 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) Specificate 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 impostate 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) Specificate il valore da utilizzare per riempire le colonne scaricate nelle colonne varchar. I valori possibili sono lo spazio o i 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) Specificate 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) Specificate 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) Specificate 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 l'applicazione gestita con il motore AWS Transform for mainframe

Puoi aggiungere un file nella `config` cartella dell'applicazione refactorizzata che ti consentirà di accedere alle nuove funzionalità del motore di runtime AWS Transform for mainframe. 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 di AWS Transform per applicazioni gestite da mainframe

Il diagramma seguente mostra dove collocare il `user-properties.yml` file all'interno della struttura dell'applicazione di esempio AWS Transform for mainframe,. 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

Opzionale. L'intervallo di commit. Blusam

Tipo: numero

Valore predefinito: 100000

card.encoding

Opzionale. Codifica delle carte: da utilizzare con. `useControlMVariable`

Tipo: stringa

Predefinito: CP1145

dimensione del file di input

Opzionale. 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

Opzionale. Speciifica 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. onInvalidNumericDati

Opzionale. Come reagire quando si decodificano dati numerici non validi. I valori consentiti sono`reject`,,`toleratespaces`. `toleratespaceslowvalues` `toleratemost`

Tipo: stringa

Impostazione predefinita: rifiutare

defaultKeepExistingFile

Opzionale. Speciifica se impostare il valore precedente predefinito del set di dati.

Tipo: booleano

Impostazione predefinita: false

disposition.checkexistence

Opzionale. Specifica se rilasciare un check on file existence per Dataset with DISP SHR o OLD.

Tipo: booleano

Impostazione predefinita: false

ExternalSort.Threshold

Opzionale. La soglia di ordinamento: determina quando il sistema deve passare dall'ordinamento in memoria a quello di unione esterna (utilizzando file temporanei nel file system). Se non viene specificata (default=`null`), la soglia viene impostata sulla metà della memoria disponibile per ogni nuova operazione di ordinamento. Per stimare la memoria disponibile, viene eseguita una chiamata a System.gc ().

Tipo: dimensione dei dati

Di default: null

```
externalSort.threshold: 12MB
```

blockSizeDefault

Opzionale. La dimensione del blocco predefinita da utilizzare per i byte BDW.

Tipo: numero

Valore predefinito: 32760

```
blockSizeDefault: 32760
```

Forza HR

Opzionale. Speciifica se utilizzare Human Readable SYSPRINT, sulla console o sull'output del file.

Tipo: booleano

Impostazione predefinita: false

Data forzata

Opzionale. 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

Opzionale. Blocca la data e l'ora nel database. Utilizzare solo durante lo sviluppo e il test.

Impostazione predefinita: false

```
frozenDate: false
```

ims.messages.ExtendedSize

Opzionale. Speciifica se impostare ExtendedSize sui messaggi ims.

Tipo: booleano

Impostazione predefinita: false

LockTimeout

Opzionale. 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

Opzionale. Elenco di prefissi da utilizzare per la trasformazione delle variabili ControlM. Ciascuno separato da una virgola.

Tipo: stringa

Predefinito: &, @,%

interrogazione. useConcatCondition

Opzionale. Specifica se la condizione della chiave è creata dalla concatenazione di chiavi o meno.

Tipo: booleano

Impostazione predefinita: false

RollbackOnRTE

Opzionale. Specifica se eseguire il rollback della transazione implicita dell'unità di esecuzione sulle eccezioni di runtime.

Tipo: booleano

Impostazione predefinita: false

sctThreadLimit

Opzionale. Il limite dei thread per l'attivazione degli script.

Tipo: numero

Impostazione predefinita: 5

sqlCodePointSpostamento

Opzionale. 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

Opzionale. Speciifica 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

Opzionale. Speciifica se alzare un abend se un passaggio fallisce o completa l'esecuzione.

Tipo: booleano

Impostazione predefinita: true

stopExecutionWhenProgNotFound

Opzionale. Speciifica 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

system.date.format

Opzionale. Specifica il formato della data di sistema.

Tipo: stringa

Impostazione predefinita: MDY

system.date.separator

Opzionale. Specifica il separatore di data del sistema.

Tipo: stringa

Impostazione predefinita: /

uppercaseUserInput

Opzionale. Speciifica se l'input dell'utente deve essere in lettere maiuscole.

Tipo: booleano

Impostazione predefinita: true

Usa Control MVariable

Opzionale. Specifica se utilizzare la specifica Control-M per la sostituzione delle variabili.

Tipo: booleano

Impostazione predefinita: false

jcl.checkpoint.expireTimeout

Opzionale. Speciifica la durata del periodo di conservazione dei checkpoint JCL nel provider di persistenza o nel registro in memoria.

Tipo: numero

Predefinito: -1

jcl.checkpoint. expireTimeoutUnit

Opzionale. Specificate l'unità di durata temporale per la `jcl.checkpoint.expireTimeout` proprietà. Valori costanti enum supportati:. `java.util.concurrent.TimeUnit`

Tipo: stringa

Impostazione predefinita: SECONDS

proprietà batchscript di Gapwalk

encoding

Opzionale. 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

Opzionale. 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

Opzionale. Elenco di comandi da disattivare, separati da una virgola. I valori consentiti sono `PGM_BASIC` `RCVMSGSNDRCVF,CHGVAR,QCLRDTAQ,RTVJOBA,ADDLFM,ADDPFM,RCVF,OVRDBF,DLTOVR,CPYF,SNDDTAQ` Utile quando si desidera disabilitare o sovrascrivere un programma esistente. `PGM_BASIC`è un programma AWS Transform for mainframe Runtime specifico progettato per scopi di debug.

Tipo: stringa

Di default: null

spring.datasource.primary.jndi-name

Opzionale. L'origine dati principale di Java Naming And Directory Interface (JNDI).

Tipo: stringa

Impostazione predefinita: jdbc/primary

Modalità suddivisa in zone

Opzionale. 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

Opzionale. La codifica dei file CL. Prevede una codifica validaCP1047,,IBM930... ASCII UTF-8

Tipo: stringa

Predefinito: CP297

cl.ZonedMode

Opzionale. La modalità per la codifica o la decodifica dei comandi del linguaggio di controllo (CL). I valori consentiti sonoEBCDIC_STRICT//EBCDIC_MODIFIED. AS400

Tipo: stringa

Impostazione predefinita: EBCDIC_STRICT

Proprietà Gapwalk JHDB

ims.programs

Opzionale. 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

Opzionale. In caso contrariojhdb.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. none 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

Opzionale. La modalità di persistenza del checkpoint. I valori consentiti sononone//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

Opzionale. La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida,,,... `CP1047` `IBM930` `ASCII` `UTF-8`

Tipo: stringa

Predefinito: CP297

jhdb. identificationCardData

Opzionale. Utilizzato per codificare alcuni «dati della carta di identificazione dell'operatore» nel campo MID indicato dal parametro CARD.

Tipo: stringa

Default: ""

jhdb.lterm

Opzionale. 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

Opzionale. La durata della cache (in millisecondi) utilizzata nella navigazione gerarchica per un RDBMS.

Tipo: numero

Impostazione predefinita: 5000

jhdb.query. limitJoinUsage

Opzionale. Specifica se utilizzare il parametro limit join usage sui grafici RDBMS.

Tipo: booleano

Impostazione predefinita: true

jhdb. use-db-prefix

Opzionale. Speciifica se abilitare un prefisso di database nella navigazione gerarchica per un RDBMS.

Tipo: booleano

Impostazione predefinita: true

Proprietà Gapwalk JICS

jics.data. dataJsonInitUbicazione

Opzionale. Posizione del file json preparato dall'Analyzer dall'analisi di CSD e utilizzato per inizializzare il database jics,

Tipo: stringa

Default: ""

jics.db. dataScriptLocation

Opzionale. Posizione dello script initJics.sql, preparato da Analyzer dall'analisi delle esportazioni CSD dal mainframe.

Tipo: stringa

Default: ""

jics.db. dataTestQueryUbicazione

Opzionale. 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

Opzionale. 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

Opzionale. 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

Opzionale. Specifica se attivare il run unit launcher pool in JICS.

Tipo: booleano

Impostazione predefinita: false

jics. runUnitLauncherDimensioni della piscina

Opzionale. 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

Opzionale. 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

Predefinito: eu-west-1

jics.xa.agent.timeout

Opzionale. 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

Opzionale. Il Regione AWS per il servizio Amazon SQS MQ.

Tipo: stringa

Predefinito: eu-west-3

TaskExecutor. allowCoreThreadTimeOut

Opzionale. 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

Opzionale. 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

Opzionale. 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

Opzionale. 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

Opzionale. Specifica se memorizzare nella cache i metadati del database.

Tipo: booleano

Impostazione predefinita: true

check-groovy-file

Opzionale. Specifica se controllare il contenuto dei file Groovy prima della registrazione.

Tipo: booleano

Impostazione predefinita: true

Statistiche del database

Opzionale. Specifica se consentire ai costruttori di SQL di raccogliere e visualizzare informazioni statistiche.

Tipo: booleano

Impostazione predefinita: false

dateTimeFormat

Opzionale. dateTimeFormat Descrive come trasferire la data, l'ora e il tipo di timestamp del database in entità semplificatrici di dati. I valori consentiti sono///ISOEURUSALOCAL

Tipo: stringa

Impostazione predefinita: ISO

dbDateFormat

Opzionale. Il formato della data di destinazione del database.

Tipo: stringa

Predefinito: yyyy-MM-dd

dbTimeFormat

Opzionale. Il formato dell'ora di destinazione del database.

Tipo: stringa

Predefinito: HH:mm:ss

dbTimestampFormat

Opzionale. Il formato del timestamp di destinazione del database.

Tipo: stringa

Predefinito: HH:mm:ss.ssssss yyyy-MM-dd

Recupera dimensione

Opzionale. Il valore fetchSize per i cursori. Da utilizzare quando si recuperano dati utilizzando chunks by utils. load/unload

Tipo: numero

Impostazione predefinita: 10

Disabilita forzatamente SQLTrim StringType

Opzionale. Specifica se disabilitare il taglio di tutti i parametri della stringa sql.

Tipo: booleano

Impostazione predefinita: false

localDateFormat

Opzionale. Elenco dei formati di data locali. Separa ogni formato con |.

Tipo: stringa

localTimeFormat

Opzionale. Elenco dei formati dell'ora locale. Separa ogni formato con |.

Tipo: stringa

localTimestampFormat

Opzionale. Elenco dei formati di timestamp locali. Separa ogni formato con. |

Tipo: stringa

Impostazione predefinita:

pgmDateFormat

Opzionale. Il formato di data e ora utilizzato nei programmi.

Tipo: stringa

Impostazione predefinita: yyyy-MM-dd

pgmTimeFormat

Opzionale. Il formato dell'ora usato per l'esecuzione di pgm (programmi).

Tipo: stringa

Predefinito: hh.mm.ss

pgmTimestampFormat

Opzionale. Il formato del timestamp.

Tipo: stringa

Impostazione predefinita: .mm.ss.ssssss yyyy-MM-dd-HH

Proprietà del programma di utilità Gapwalk

jcl.type

Facoltativo. `.jcl`tipo di file. I valori consentiti sono`jcl`/`vse`. I PRINT/REPRO comandi dell'utilità IDCAMS restituiscono 4 se il file è vuoto per jcl non vse.

Tipo: stringa

Predefinito: mvs

listcat.variablelengthpreprocessor.enabled

Opzionale. Speciifica se abilitare il preprocessore a lunghezza variabile per il comando LISTCAT.

Tipo: booleano

Impostazione predefinita: false

listcat.variablelengthpreprocessor.type

Opzionale. Il tipo di oggetti contenuti nel file listcat, se abilitato. `listcat.variablelengthpreprocessor.enabled` I valori consentiti sono`rdw/bdw`.

Tipo: stringa

Impostazione predefinita: rdw

load.batchSize

Opzionale. La dimensione del batch dell'utilità di caricamento.

Tipo: numero

Impostazione predefinita: 0

load.format.dbdate

Opzionale. Il formato del database dell'utilità di caricamento da utilizzare.

Tipo: stringa

Impostazione predefinita: yyyy-MM-dd

Load.format.dbTime

Opzionale. Il tempo di utilizzo del database dell'utilità di caricamento.

Tipo: stringa

Impostazione predefinita: HH:mm:ss

load.format.localDate

Opzionale. Il formato di data locale dell'utilità di caricamento da utilizzare.

Tipo: stringa

Impostazione predefinita: gd.mm.yyyy| |yyyy-MM-DD dd/MM/yyyy

load.format.localTime

Opzionale. Il formato dell'ora locale dell'utilità di caricamento da utilizzare.

Tipo: stringa

Predefinito: HH:mm:ss|Hh.mm.ss

caricare. sqlCodePointCambio

Opzionale. L'utilità pointshift del codice SQL per il caricamento. Esegue il processo di cambio dei caratteri. Richiesto quando il database di destinazione DB2 è Postgresql.

Tipo: numero

Impostazione predefinita: 0

sysPunchEncoding

Opzionale. Il set di caratteri di codifica syspunch. I valori supportati sono/. `Cp1047` `ASCII`

Tipo: stringa

Impostazione predefinita: ASCII

treatLargeNumberAsInteger

Opzionale. Speciifica se trattare i numeri grandi come. `Integer` Per impostazione predefinita, vengono `BigDecimal` trattati come tali.

Tipo: booleano

Impostazione predefinita: false

unload.chunksize

Opzionale. Dimensione del blocco utilizzata per l'utilità di scarico.

Tipo: numero

Impostazione predefinita: 0

unload.columnFiller

Opzionale. Il riempitore di colonne dell'utilità di scarico.

Tipo: stringa

Impostazione predefinita: spazio

unload.fetchSize

Opzionale. Consente di regolare la dimensione del recupero quando si gestiscono i cursori nell'utilità unload.

Tipo: numero

Impostazione predefinita: 0

unload.format.date

Opzionale. Se abilitato, `unload.useDatabaseConfiguration` il formato della data da utilizzare nell'utilità di scarico.

Tipo: stringa

Impostazione predefinita: MM/dd/yyyy

unload.format.time

Opzionale. Se abilitato, `unload.useDatabaseConfiguration` il formato dell'ora da utilizzare nell'utilità di scarico.

Tipo: stringa

Predefinito: hh.mm.ss

unload.format.timestamp

Opzionale. Se abilitato, `unload.useDatabaseConfiguration` il formato del timestamp da utilizzare nell'utilità unload.

Tipo: stringa

Impostazione predefinita: .mm.ss.ssssss yyyy-MM-dd-HH

scarica.nbi. whenNotNull

Opzionale. Il valore Null Byte Indicator (nbi) da aggiungere quando il valore del database non è nullo.

Tipo: esadecimale

Valore predefinito: 00

unload.nbi.whenNull

Opzionale. Il valore dell'indicatore Null Byte (nbi) da aggiungere quando il valore del database è nullo.

Tipo: esadecimale

Predefinito: 6F

scarica.nbi. writeNullIndicator

Opzionale. Specifica se scrivere l'indicatore null nel file di output di scaricamento.

Tipo: booleano

Impostazione predefinita: false

scaricare. sqlCodePointCambio

Opzionale. L'utilità pointshift del codice SQL per unload. Esegue il processo di cambio dei caratteri. Richiesto quando il database di destinazione DB2 è Postgresql.

Tipo: numero

Impostazione predefinita: 0

scaricare. useDatabaseConfiguration

Opzionale. Specifica se utilizzare la configurazione di data o ora da application-main.yml nell'utilità unload.

Tipo: booleano

Impostazione predefinita: false

scaricare. varCharIsNullo

Opzionale. 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

Opzionale. Per specificare quando è abilitato. `qtemp.dblog` La durata della partizione db (in ore).

Tipo: numero

Impostazione predefinita: 0

qtemp.dblog

Opzionale. Se abilitare la registrazione del database QTEMP.

Tipo: booleano

Impostazione predefinita: false

qtemp.uuid.length

Opzionale. La lunghezza dell'ID univoco di QTEMP.

Tipo: numero

Valore predefinito: 9

quartz.scheduler. stand-by-if-error

Opzionale. Speciifica 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

Opzionale. Speciifica se caricare tutti i dati della tabella Datacom in una cache di riscaldamento all'avvio del server.

Tipo: booleano

Impostazione predefinita: false

Modernizzazione del mainframe AWS riferimento alla definizione dell'applicazione

In Modernizzazione del mainframe AWS, si configurano 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 Transform for mainframe e Rocket Software (precedentemente Micro Focus) e identifica tutti gli elementi obbligatori e opzionali.

Indice

- Sezione generale dell'intestazione
- Panoramica della sezione relativa alle definizioni
- AWS Esempio di definizione dell'applicazione Transform for mainframe
- AWS Transform for mainframe: dettagli della definizione
  - Listener (i): obbligatorio
  - AWS Transform per l'applicazione mainframe: obbligatorio
  - Blusam- opzionale
  - AWS Trasforma per le code di messaggi del mainframe: opzionale
  - AWS Transform for mainframe Application Storage, configurazione EFS - opzionale
- Definizione dell'applicazione Rocket Software (precedentemente Micro Focus)
- Dettagli della definizione di Rocket Software
  - Listener (i): obbligatorio
  - Posizioni dei set di dati: obbligatorie
  - Gestore di autenticazione e autorizzazione Amazon Cognito: opzionale
  - Gestore LDAP e Active Directory: opzionale
  - Impostazioni Batch: obbligatorie
  - Impostazioni CICS: obbligatorie
  - Stampanti: opzionali
  - Risorse XA: opzionali
  - Impostazioni di runtime: opzionali

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.

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",
                "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",
              "s3-key-prefix": "v1"
          }
      }
  ]
```

versione modello

(Obbligatorio) Specificate 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

Speciifica le posizioni dei file e delle altre risorse richieste dall'applicazione durante l'esecuzione.

source-id

Specificate un nome per la posizione. Questo nome viene utilizzato per fare riferimento alla posizione di origine in base alle esigenze nella definizione dell'applicazione JSON.

tipo di sorgente

Speciifica il tipo di fonte. Attualmente, l'unico valore consentito è s3.

properties

Fornisce i dettagli della posizione di origine. Ogni proprietà viene specificata con una stringa.

- `s3-bucket`- Obbligatorio. Speciifica il nome del bucket Amazon S3 in cui sono archiviati i file.
- `s3-key-prefix`- Obbligatorio. Speciifica il nome della cartella nel bucket Amazon S3 in cui sono archiviati i file.

Panoramica della sezione relativa alle definizioni

Speciifica 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, Modernizzazione del mainframe AWS 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 Transform for mainframe

```
{
 "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 Transform for mainframe: dettagli della definizione

Listener (i): obbligatorio

Specificate la porta che utilizzerete per accedere all'applicazione tramite Elastic Load Balancing Modernizzazione del mainframe AWS creato. 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. Assicurati che non ci siano altri listener o applicazioni che operano su questa porta.

tipo

(Obbligatorio) Attualmente, `http` è supportato solo.

AWS Transform per l'applicazione mainframe: obbligatorio

Specificate la posizione in cui il motore preleva il file di immagine dell'applicazione utilizzando la seguente struttura.

```
"ba-application": {
            "app-location": "${s3-source}/murachs-v6/",
            "files-directory": "/m2/mount/myfolder",
            "enable-jics": <true|false>,
            "enable-batch-restart": <true|false>,
            "shared-app-location": "${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 input/output file per i batch. Deve essere una sottocartella della configurazione di Amazon EFS o Amazon FSx mount point a livello di ambiente. La sottocartella deve essere di proprietà di un utente idoneo per essere utilizzata dall'applicazione AWS Transform for mainframe in esecuzione all'interno. Modernizzazione del mainframe AWS A tal fine, quando si collega l'unità a un'istanza Linux Amazon EC2, è necessario creare un gruppo con `101` ID e un utente con `3001` ID e la cartella desiderata deve essere di proprietà di questo utente. Ad esempio, in questo modo, la *`testclient`* cartella può essere utilizzata da AWS Transform for mainframe Modernizzazione del mainframe AWS Managed.

```
groupadd -g 101 mygroup
useradd -M -g mygroup -p mypassword -u 3001 myuser
mkdir testclient
chown myuser:mygroup testclient
```

abilita jics

(Facoltativo) Specifica se abilitare JICS. Il valore predefinito è true. L'impostazione su false impedisce la generazione del database JICS.

enable-batch-restart

(Facoltativo) Specifica se abilitare la funzionalità di riavvio per i lavori in batch. Il valore predefinito è false (falso). Per ulteriori informazioni sulle configurazioni di riavvio in batch, vedere AWS

Transform for mainframe Engine con il prefisso `jcl.checkpoint` Proprietà di configurazione per l'applicazione gestita con AWS Transform for mainframe engine.

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 Blusam database 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) Speciifica quanti thread dedicati vengono utilizzati per il meccanismo di scrittura dietro a cui si basa il Blusam motore. Il valore di default è 8.
- `batch-size`- (Facoltativo) Specifica 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 della dimensione

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 Gestione dei segreti AWS 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 false`useSsl`.
- `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 Gestione dei segreti AWS database segreto.

AWS Trasforma per le code di messaggi del mainframe: opzionale

Specificare i dettagli della connessione JMS-MQ per l'applicazione AWS Transform for mainframe.

```
"message-queues": [
  {
      "product-type": "JMS-MQ",
      "queue-manager": "QMgr1",
      "channel": "mqChannel1",
      "hostname": "mqserver-host1",
      "port": 1414,
      "user-id": "app-user1",
      "ssl-cipher": "*TLS12ORHIGHER",
      "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",
      "ssl-cipher": "*TLS12ORHIGHER",
      "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  }
]
```

tipo di prodotto

(Obbligatorio) Specificare il tipo di prodotto. Attualmente, questo può essere solo «JMS-MQ» per AWS Transform for mainframe.

gestore di code

(Obbligatorio) Specificate il nome del gestore delle code.

channel

(Obbligatorio) Specificate il nome del canale di connessione al server.

hostname

(Obbligatorio) Specificate il nome host del server della coda dei messaggi.

port

(Obbligatorio) Specificate il numero di porta del listener su cui il server è in ascolto.

ID utente

(Facoltativo) Speciifica l'ID dell'account utente autorizzato a eseguire operazioni di coda dei messaggi sul canale specificato.

cifrario ssl

(Facoltativo) Specifica la specifica del cifrario SSL per la connessione.

secret-manager-arn

(Facoltativo) Specifica l'Amazon Resource Name (ARN) di Secrets Manager che fornisce la password dell'utente specificato.

AWS Transform for mainframe Application Storage, configurazione EFS - opzionale

Specificare i dettagli del punto di accesso EFS per lo storage delle applicazioni utilizzando la seguente struttura.

```
"ba-application": {
            "file-permission-mask": "UMASK002"
},
"efs-configs": [
           {
            "file-system-id": "fs-01376dfsvfvrsvsr",
            "mount-point": "/m2/mount/efs-ap2",
            "access-point-id": fsap-0eaesefvrefrewgv8"
           }
]
```

file-system-id

(Obbligatorio) L'ID del file system EFS a cui si riferisce il punto di accesso. Modello: «fs- ([0-9a-f] {8,40}) {1,128} $»

punto di montaggio

(Obbligatorio) Il punto di montaggio per il file system a livello di applicazione. Deve essere diverso dal punto di montaggio dello storage a livello di ambiente.

access-point-id

(Obbligatorio) L'ID del punto di accesso, assegnato da Amazon EFS. Modello: «^fsap- ([0-9a-f] {8,40}) {1,128} $»

file-permission-mask

(Facoltativo) Definisce la maschera di creazione dei file per i file creati dal processo di applicazione. Ad esempio, quando il valore è impostato su`UMASK006`, tutti i file avranno l'autorizzazione 660. Ciò significa che solo il proprietario e il gruppo di file avranno l'accesso in lettura e scrittura, mentre gli altri utenti non dispongono di alcuna autorizzazione.

> **Note**
>
> Il valore impostato per questo campo viene considerato solo quando si utilizza lo storage EFS a livello di applicazione.

> **Note**
>
> Quando viene fornito efs config, la directory dei file deve essere specificata nella sezione di definizione dell'applicazione. Deve essere una sottocartella del punto di montaggio Amazon EFS configurato a livello di applicazione.

Definizione dell'applicazione Rocket Software (precedentemente Micro Focus)

La seguente sezione di definizione di esempio riguarda il motore di runtime Rocket Software e contiene elementi obbligatori e 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": "${s3-source}/batch/jcl",
            "program-path": "/m2/mount/libs/loadlib:$EFS_MOUNT/emergency/loadlib",
            "system-procedure-libraries":"SYS1.PROCLIB;SYS2.PROCLIB",
            "aliases": [
               {"alias": "FDSSORT", "program": "SORT"},
               {"alias": "MFADRDSU", "program": "ADRDSSU"}
            ]
        },
        "cics-settings": {
            "binary-file-location": "${s3-source}/cics/binaries",
            "csd-file-location": "${s3-source}/cics/def",
            "system-initialization-table": "BNKCICV"
        },
        "jes-printers": [
           {
             "name": "printerName",
              "classes": [
                 "A",
                 "B"
             ],
             "description": "printer desc....",
             "exit-module": {
                 "name": "lrsprte6"
                }
             }
         ],
        "xa-resources" : [{
            "name": "XASQL",
            "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
            "xa-connection-type": "postgres",
            "module": "${s3-source}/xa/ESPGSQLXA64.so"
        }],
        "runtime-settings": {
          "base-configuration-location": "${s3-source}/exported.json",
          "environment-variables": {
            "ES_JES_RESTART": "N",
```

```
            "EFS_MOUNT": "/m2/mount/efs",
            "LRSQ_ADDRESS": "<lrsq-address>"
          }
        }
    }
}
```

Dettagli della definizione di Rocket Software

Il contenuto nella sezione delle definizioni del file di definizione dell'applicazione Rocket Software 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.

tipo

Speciifica 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

Speciifica la posizione dei set di dati creati dall'applicazione migrata. Attualmente Modernizzazione del mainframe AWS supporta solo set di dati provenienti da un singolo database VSAM.

- `name`- Speciifica il nome dell'istanza del database che contiene i set di dati creati dall'applicazione migrata.
- `secret-manager-arn`- Speciifica l'Amazon Resource Name (ARN) del segreto che contiene le credenziali del database.

Gestore di autenticazione e autorizzazione Amazon Cognito: opzionale

Modernizzazione del mainframe AWS utilizza Amazon Cognito per l'autenticazione e l'autorizzazione per le 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

Speciifica il pool Modernizzazione del mainframe AWS di utenti di Amazon Cognito utilizzato per autenticare gli utenti dell'applicazione migrata. Il pool di utenti Regione AWS per l'applicazione deve corrispondere a quello per l' Regione AWS applicazione. Modernizzazione del mainframe AWS

identificativo del cliente

Speciifica l'applicazione migrata a cui l'utente autenticato può accedere.

identity-pool-id

Speciifica 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. Modernizzazione del

mainframe AWS Il pool di identità Regione AWS per il pool di identità deve corrispondere a quello per l'applicazione Regione AWS . Modernizzazione del mainframe 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 Gestione dei segreti AWS segreto con AD/LDAP i tuoi dati 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 Gestione dei segreti AWS 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`, Modernizzazione del mainframe AWS 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 che non dispone di più autorizzazioni rispetto alle autorizzazioni di sola lettura e verifica più restrittive necessarie per l'esecuzione dell'applicazione. `authorizedId`
- Ti consigliamo di ruotare le credenziali regolarmente. AD/LDAP

- Non creare utenti AD con il nome utente `awsuser` o. `mfuser` Questi due nomi utente sono riservati all' AWS uso.

Di seguito è riportato un esempio di :

```
{
    "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 Modernizzazione del mainframe AWS le `DescribeSecret` autorizzazioni `GetSecretValue` e per il segreto e le autorizzazioni per la `Decrypt` chiave `DescribeKey` KMS. Per ulteriori informazioni, consulta Autorizzazioni per la chiave KMS nella Guida per l'utente. Gestione dei segreti AWS

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 di :

```
"ldap-ad-auth-handler": {
    "ldap-ad-connection-secrets": ["arn:aws:secrets:1234:us-east-1:secret:123456"]
}
```

Se l'applicazione è integrata con LDAP ed è stata avviata, è necessario fornire le credenziali per eseguire almeno un'operazione relativa all'applicazione menzionata nell'elenco delle autorizzazioni supportate.

Il gestore di LDAP/AD autenticazione è disponibile per Micro Focus (Rocket) 8.0.11 e versioni successive.

> **Note**
>
> Attualmente, l'amministratore LDAP deve fornire le autorizzazioni di «modifica» sull'`casstart`utilità nelle risorse del server aziendale «OPERCMDS» nella propria directory LDAP. Questa operazione deve essere eseguita per tutti gli utenti predefiniti richiesti (ad esempio, CICSUSER, se l'applicazione è correlata a CICS) per avviare l'applicazione con successo.

Fornire credenziali utente LDAP per l'autenticazione e l'autorizzazione

1. Crea un file Gestione dei segreti AWS con le seguenti chiavi e valori:

```
{
  "username" : "<USERNAME>",
  "password" : "<PASSWORD>"
}
```

> **Important**
>
> Dovresti avere i diritti di esecuzione `DescribeSecrets` e `GetSecretValue` di utilizzo del Secrets Manager. Inoltre, associa una chiave KMS e le autorizzazioni necessarie per Gestione dei segreti AWS, come indicato in Scelta di un. AWS KMS key

2. Scegli il parametro Secrets Manager.

   AWS console

   Durante l'esecuzione delle operazioni dalla AWS console, ci sarà un'opzione per scegliere il Secrets Manager che deve essere passato.

   AWS CLI (or SDK)

   Durante l'esecuzione di operazioni da AWS CLI (o SDK), il parametro API `auth-secrets-manager-arn` deve essere passato con Secrets Manager ARN.

Ecco l'elenco delle operazioni applicative che attualmente supportano l'autorizzazione:

- `StartBatchJob`
- `CancelBatchJobExecution`
- `ListBatchJobRestartPoints`

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": "${s3-source}/batch/jcl",
            "program-path": "/m2/mount/libs/loadlib:$EFS_MOUNT/emergency/loadlib",
            "system-procedure-libraries":"SYS1.PROCLIB;SYS2.PROCLIB",
            "aliases": [
               {"alias": "FDSSORT", "program": "SORT"},
               {"alias": "MFADRDSU", "program": "ADRDSSU"}
    ]
}
```

iniziatori

Speciifica 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. Esempio:

```
"batch-settings": {
            "initiators": [
                {
                  "classes": ["A", "B"],
                  "description": "initiator...."
                },
                {
                  "classes": ["C", "D"],
                  "description": "initiator...."
                }
```

```
        ],
}
```

Per ulteriori informazioni, vedere Per definire un iniziatore batch o una stampante SEP nella documentazione di Micro Focus Enterprise Server.

- `classes`- Speciifica 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

Speciifica la posizione dei file JCL (Job Control Language) richiesti dai processi batch eseguiti dall'applicazione migrata.

percorso del programma

Speciifica il percorso richiesto per eseguire i processi batch quando un programma in un JCL non si trova nella posizione predefinita. I diversi nomi di percorso sono separati da due punti (:).

> **Note**
>
> Il percorso del programma può essere solo un percorso EFS.

system-procedure-libraries

Speciifica i set di dati partizionati predefiniti in cui verranno cercate le procedure JCL. La procedura, tuttavia, non si trova nel JCL o tramite le istruzioni JCLLIB. Questi set di dati devono essere catalogati e deve essere utilizzato il nome del catalogo. E le voci sono separate da un punto e virgola (;).

alias

Definisce una mappatura per i nomi delle utilità e dei programmi utilizzati in JCL con il nome di implementazione dell'utilità. AWS e le utilità batch di terze parti (ad esempio M2SFTP, M2WAIT, Syncsort, ecc.) possono opzionalmente avere alias per eliminare la necessità di modificare il JCL. Esempio:

- FDSSORT Alias FDSSORT per SORT e Alias FDSICET per ICETOOL
- Alias ADRDSSU MFADRDSU per ADRDSSU

- Alias Syncsort DMXMFSRT per SORT

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

Speciifica la posizione dei file del programma di transazione CICS.

csd-file-location

Speciifica 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

Speciifica 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.

Stampanti: opzionali

Specificate una stampante jes utilizzando la seguente struttura.

```
"jes-printers": [
    {
        "name": "printerName",
        "classes": [
            "A",
            "B"
        ],
```

```
        "description": "printer desc....",
        "exit-module": {
            "name": "lrsprte6",
            "module" : "program"
        }
    }
],
```

> **Note**
>
> È possibile configurare un massimo di 25 stampanti per una determinata applicazione.

nome

(Obbligatorio) Specificate il nome da associare a questa risorsa di stampa. I nomi devono essere univoci per ogni stampante ed è possibile utilizzare un limite di 128 caratteri alfanumerici.

classi

(Obbligatorio) Specificate le classi di output applicabili a questa risorsa di stampa. È possibile utilizzare un limite di 36 caratteri alfanumerici.

description

(Facoltativo) Testo descrittivo aggiuntivo per la stampante.

modulo di uscita

(Facoltativo) Specifica un modulo personalizzato per l'uscita dalla stampa. Non ci sono valori predefiniti, se non viene specificato, non verrà utilizzato alcun modulo di uscita. È possibile utilizzare un modulo di uscita dalla stampa gestito o fornire il proprio. I moduli di uscita dalla stampa gestita vengono definiti utilizzando il nome riservato `lrsprte6` per la coda LRS o forniti da soli utilizzando il parametro del modulo per specificare la posizione e il nome.

La struttura di `exit-module` ha due componenti:

- `name`- (Obbligatorio), se `exit-module` utilizzato. Il nome della voce del modulo di uscita. Il nome di immissione del modulo di uscita è limitato a un massimo di 8 caratteri.
- `module`- (Facoltativo) La posizione S3 del binario del modulo di uscita di stampa.

Puoi vedere altri esempi di definizione del modulo di uscita nella the section called "Stampanti" sezione.

Risorse XA: opzionali

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",
            "xa-connection-type": "postgres",
            "module": "${s3-source}/xa/ESPGSQLXA64.so"
}]
```

> **Note**
>
> La definizione delle risorse XA è stata aggiornata per includere un campo opzionale`xa-connection-type`. Se non viene fornito, si presume che il tipo di connessione sia «postgres».

nome

(Obbligatorio) Specificate il nome della risorsa XA.

secret-manager-arn

(Obbligatorio) Specifica l'Amazon Resource Name (ARN) per il segreto che contiene le credenziali per la connessione al database.

xa-connection-type

(Facoltativo) Speciifica il tipo di connessione alla risorsa XA.

modulo

(Obbligatorio) Specificate la posizione del file eseguibile del modulo di commutazione RM. Per ulteriori informazioni, vedere Pianificazione e progettazione XARs nella documentazione di Micro Focus Enterprise Server.

Impostazioni di runtime: opzionali

Specificate i dettagli richiesti per le impostazioni di runtime per gestire le variabili di ambiente consentite utilizzando la seguente struttura.

```
"runtime-settings": {
          "base-configuration-location": "${s3-source}/exported.json",
          "environment-variables": {
            "ES_JES_RESTART": "N",
            "EFS_MOUNT": "/m2/mount/efs"
    }
}
```

base-configuration-location

(Facoltativo) Speciifica la posizione per l'importazione in blocco per una configurazione del server Micro Focus. Questo file deve essere un JSON valido e deve trovarsi nella stessa posizione S3 della posizione degli artefatti dell'applicazione definita sopra. Per esportare la configurazione da un'applicazione esistente, consulta la sezione Esportare una regione da ESCWA nella documentazione del software Rocket.

variabili d'ambiente

Specificano le variabili di ambiente supportate da Micro Focus che vengono applicate al runtime di questa applicazione.

- `ES_JES_RESTART`è una variabile di ambiente Rocket Software che consente l'elaborazione di riavvio di JCL. Facoltativamente, puoi anche utilizzarla `ES_ALLOC_OVERRIDE` come variabile di ambiente Rocket Software.
- `EFS_MOUNT`è una variabile di ambiente personalizzata che l'applicazione potrebbe utilizzare per identificare dove si trova il mount EFS dell'ambiente.

Puoi accedere a tutte le variabili di ambiente di Rocket Software nella guida Rocket Enterprise Server for UNIX.

AWS Riferimento alla definizione del set di dati per la modernizzazione del mainframe

Se l'applicazione richiede più di pochi set di dati per l'elaborazione, inserirli uno per uno nella console di modernizzazione del AWS mainframe è inefficiente. 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 I set di dati devono essere in un formato che possa essere caricato nel motore di runtime selezionato. In molti casi può trattarsi di un file sequenziale, ma per Rocket Software (precedentemente Micro Focus) VSAM dovrà essere nel formato proprietario. L'`DFCONV`utilità è il metodo consigliato per convertire il file. Specificare il nome del bucket e della cartella nel file JSON di definizione del set di dati.
> Per ulteriori informazioni sul motore di runtime di Rocket Software, consulta DFCONV Batch File Conversion nella documentazione del software Rocket.
> Per ulteriori informazioni su AWS Transform for mainframe, consulta. the section called "AWS Configurazione Transform for mainframe Runtime"

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 le generazioni PS o GDG
- 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 state importando, 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. (`storageTypeerelativePath`).
- 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. Speciifica 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 Transform for mainframe: i file system non sono supportati. È necessario utilizzare un database.
- Motore di runtime Rocket Software: sono supportati sia i database che i file system. Puoi utilizzare Amazon Relational Database Service o Amazon Aurora per i database e Amazon Elastic File System o FSx Amazon 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. Speciifica la posizione relativa del set di dati nel database o nel file system.

DataSet.org

(Obbligatorio) Specificate il tipo di set di dati. I valori possibili sono `vsam`, `gdg`, `ps`, `po` o `unknown`.

- AWS Motore di runtime Transform for mainframe: sono supportati solo set di dati di tipo VSAM.

- Motore di runtime Rocket Software: sono supportati i 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

- AWS Motore di runtime Transform for mainframe: supportato.
- Motore di runtime Rocket Software: 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) Specificate la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (`A`), EBCDIC () e Unknown (`E`). `?`

format

(Obbligatorio) Specificate il tipo di set di dati VSAM e il formato del record.

- AWS Transform for mainframe runtime engine: i valori possibili sono ESDS (`ES`) e KSDS (). `KS` Il formato di registrazione può essere fisso o variabile.
- Motore di runtime Rocket Software: i valori possibili sono ESDS (`ES`), KSDS (`KS`) e RRDS (). `RR` La definizione VSAM include il formato di registrazione, quindi non è necessario specificarlo separatamente.

primaryKey

(Obbligatorio) 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) Specificate la lunghezza di un record. Per i formati di record a lunghezza fissa, questi valori devono corrispondere.

- AWS Transform for mainframe runtime engine: per VSAM ESDS e KSDS, è facoltativo ed è obbligatorio. `min` `max`
- Motore di runtime Rocket Software: e sono obbligatori. `min` `max`

Ubicazione esterna

(Obbligatorio) Specifica la posizione di origine, ovvero il bucket Amazon S3 in cui hai caricato il set di dati.

Proprietà specifiche del motore di AWS Transform per mainframe

Il motore di runtime AWS Transform for mainframe 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) Speciifica 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

- AWS Motore di runtime Transform for mainframe: non supportato.
- Motore di runtime Rocket Software: 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 le generazioni PS o GDG

- AWS Motore di runtime Transform for mainframe: non supportato.
- Motore di runtime Rocket Software: supportato.

Se stai importando set di dati di generazioni PS o GDG, specifica `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) Specificate il formato dei record del set di dati. I valori possibili sono `F`, `FA`, `FB`, `FBA`, `FBM`, `FBS`, `FM`, `FS`, `LSEQ`, `U`, `V`, `VA`, `VB`, `VBA`, `VBM`, `VBS`, `VM` e `VS`.

encoding

(Obbligatorio) Specificate la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (`A`), EBCDIC () e Unknown (`E`) ?

Lunghezza del record

(Obbligatorio) Specificate 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 codice 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) Specificate il formato dei record del set di dati. I valori possibili sono `F`, `FA`, `FB`, `FBA`, `FBM`, `FBS`, `FM`, `FS`, `LSEQ`, `U`, `V`, `VA`, `VB`, `VBA`, `VBM`, `VBS`, `VM` e `VS`.

encoding

(Obbligatorio) Specificate la codifica del set di caratteri del set di dati. I valori possibili sono ASCII (`A`), EBCDIC () e Unknown (`E`). ?

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) Specificate 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 Rocket Software 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 per la modernizzazione del AWS mainframe
- Inizia con la modernizzazione del AWS mainframe
- Tutorial: configurare il runtime gestito per AWS Transform for mainframe
- Tutorial: Configurazione del runtime gestito per Rocket Software (precedentemente 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 Transform for mainframe se utilizzate il pattern di refactoring automatizzato e Rocket Software (precedentemente 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 runtime di modernizzazione AWS 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

Utilizzate 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 per la 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/
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, immettere 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 automatico o Micro Focus (Rocket) per il replatforming.
    d. Scegliete una versione per il motore che avete 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. Scegli Next (Successivo).
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 il tipo di rete. Se lo desideri IPv4, le applicazioni dell'ambiente di modernizzazione del AWS mainframe soddisfano solo IPv4 le richieste. Nella modalità dual-stack, le applicazioni soddisferanno entrambe le richieste. IPv4 IPv6 Se scegli la modalità dual-stack, assicurati che ci sia almeno 1 VPC con sottoreti abilitate. IPv6

   c. Scegli un Virtual Private Cloud (VPC).

   d. Se utilizzi il modello di alta disponibilità, scegli due o più sottoreti. Se utilizzi il pattern autonomo con il motore AWS Transform for mainframe, scegli due o più sottoreti. Se utilizzi il pattern autonomo con il motore Rocket Software, puoi specificare una sottorete.

   e. Scegli un gruppo di sicurezza per il VPC selezionato.

   > **Note**
   >
   > AWS Mainframe Modernization crea un Network Load Balancer per distribuire le connessioni all'ambiente di runtime. Assicurati che le regole in entrata e in uscita del tuo gruppo di sicurezza consentano l'accesso da un indirizzo IP alla porta specificata nella `the section called "Listener (i): obbligatorio"` proprietà della definizione dell'applicazione. Per ulteriori informazioni, consulta Aggiornare i gruppi di sicurezza per il Network Load Balancer nella Guida dell'utente per Network Load Balancer.

f. Nel campo Chiave KMS, scegli Personalizza le impostazioni di crittografia se desideri utilizzare una chiave gestita dal cliente. AWS KMS key Per ulteriori informazioni, consulta Crittografia dei dati inattiva per il servizio di modernizzazione AWS 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

g. (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

h. Scegli Next (Successivo).

7. (Facoltativo) Nella pagina Allega storage, scegli uno o più FSx file system Amazon EFS o Amazon.

   Il file system montato in un ambiente di modernizzazione del AWS mainframe deve essere di proprietà di un utente idoneo per essere utilizzato dalle applicazioni in esecuzione nella console di modernizzazione AWS Mainframe.

   Per configurare queste impostazioni utente, puoi collegare l'unità a un'istanza Linux Amazon EC2. Quindi crea un gruppo con ID `101` e un utente con ID. `3001` Inoltre, assicurati che la cartella di dati desiderata che verrà utilizzata dalle tue applicazioni sia di proprietà di questo utente.

   Ad esempio, la `myFiles` cartella può essere utilizzata dalle applicazioni di modernizzazione del AWS mainframe in esecuzione in Mainframe Modernization AWS Managed.

```
groupadd -g 101 mygroup
useradd -M -g mygroup -p mypassword -u 3001 myuser
mkdir myFiles
chown myuser:mygroup myFiles
```

> **Note**
>
> Per consentire l'accesso al file system, è necessario configurare le seguenti regole dei gruppi di sicurezza per stabilire la connettività di rete tra l'istanza di ambiente EFS e M2:
>
> - Gruppo di sicurezza ambientale M2: include una regola in uscita che consente il traffico sulla porta NFS 2049.
> - Gruppo di sicurezza destinato al montaggio sul file system: include una regola in entrata che consente il traffico sulla porta NFS 2049 dal gruppo di sicurezza dell'istanza (elencato sopra) e una regola in uscita che consente il traffico sulla porta NFS 2049.

8. Scegli Next (Successivo).
9. 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.
   - Per specificare una finestra di manutenzione particolare, scegliete 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, consultaAWS Finestra di manutenzione della modernizzazione del mainframe.

   Scegli Next (Successivo).
10. 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 per la 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/
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. Scegli Next (Successivo).
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 due ore. Tutte le modifiche al sistema vengono applicate durante questo periodo. La finestra di manutenzione consente di

controllare quando vengono apportate le modifiche e le patch software e di sicurezza. Se è pianificato un evento di manutenzione per una determinata settimana, questo ha inizio durante quella finestra di manutenzione di due ore. La maggior parte degli eventi di manutenzione viene completata anche durante la finestra di manutenzione di due ore, anche se eventi di manutenzione più estesi potrebbero richiedere più di un paio d'ore per essere completati.

La finestra di manutenzione di due ore 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 2 ore 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.

Arresta un ambiente di runtime di modernizzazione AWS 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 per la 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.

> **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/
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. Scegli Next (Successivo).
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/
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. Scegli Next (Successivo).
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 del AWS mainframe.

Queste istruzioni presuppongono che la procedura in Configurazione per la 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.

Come eliminare un ambiente

1. Apri la console di modernizzazione del AWS mainframe all'indirizzo. 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.

Refattorizzazione automatica delle applicazioni con AWS Transform for mainframe

Il refactoring automatizzato con AWS Transform for mainframe 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 di refactoring AWS Transform for mainframe, disponibile tramite Single Sign-On dalla console di modernizzazione del AWS mainframe. Per ulteriori informazioni su AWS Transform for mainframe refactor, consulta la documentazione Transform for mainframe refactor.AWS

Quando sei soddisfatto del codice sorgente trasformato, è il momento di passare a AWS: creare e distribuire l'applicazione refattorizzata utilizzando. the section called "AWS Transform for mainframe Runtime"

Questa guida ti aiuterà a comprendere i concetti di runtime e a usarli nel tuo progetto di modernizzazione.

Argomenti

- AWS Transform per versioni mainframe
- AWS Concetti di Transform for mainframe Runtime
- Configurare la configurazione per AWS Transform for mainframe Runtime
- AWS Codici di errore di runtime Transform for mainframe
- AWS Transform for mainframe Runtime APIs
- AWS Transform per mainframe Runtime Utilities
- Configurare AWS Transform for mainframe Runtime
- Modifica il codice sorgente con AWS Transform for mainframe Developer IDE

- AWS Domande frequenti su Transform for mainframe

AWS Transform per versioni mainframe

AWS Il motore Transform for mainframe offre diverse versioni tra cui scegliere. Questa pagina offre una panoramica del funzionamento del controllo delle versioni di AWS Transform for mainframe, delle modifiche apportate a ciascuna versione, delle istruzioni per l'aggiornamento delle versioni, del modo in cui gli aggiornamenti di AWS Transform for mainframe vengono comunicati ai clienti e del ciclo di vita di queste versioni.

La the section called "AWS Transform per il controllo delle versioni mainframe" pagina fornisce informazioni dettagliate sulle versioni e su come ciascuna versione può essere identificata dalle versioni principali e secondarie. La the section called "AWS Note sulla versione di Transform for mainframe" pagina contiene note di rilascio dettagliate per ogni versione principale e secondaria. the section called "AWS Trasforma per le vulnerabilità di sicurezza del mainframe"la pagina illustra come AWS Transform for mainframe gestisce le vulnerabilità e le esposizioni comuni (CVE). the section called "Aggiornamento di Transform per mainframe AWS "descrive le istruzioni di aggiornamento per le versioni di Transform for mainframe. AWS E the section called "AWS Trasformazione per il ciclo di vita del mainframe" include tutti i dettagli sulle date di fine del ciclo di vita (EOL) per le versioni principali di AWS Transform for mainframe Runtime.

Argomenti

- AWS Transform per il controllo delle versioni mainframe
- AWS Transform for mainframe Runtime
- AWS Note sulla versione di Transform for mainframe
- AWS Trasforma per le vulnerabilità di sicurezza del mainframe
- Istruzioni per l'aggiornamento di AWS Transform for mainframe
- AWS Trasformazione per il ciclo di vita del mainframe

AWS Transform per il controllo delle versioni mainframe

Le versioni dei prodotti AWS Transform for mainframe Transformation e Runtime utilizzano uno schema conforme a semver (semantic versioning). Per distribuire l'applicazione, è necessario utilizzare la versione di runtime corrispondente compatibile con il codice modernizzato. Se avete domande sulla versione da utilizzare, contattate il vostro responsabile della distribuzione di AWS Transform for mainframe.

Rilasci

Ogni versione è identificata da uno **`[Major].[Minor].[Patch]`**schema. Ad esempio, con la versione AWS Transform for mainframe Runtime`4.1.0`, la versione principale è 4, la versione secondaria è 1 e la versione patch è 0.

Intendiamo rilasciare mensilmente nuove versioni secondarie di AWS Transform for mainframe Runtime e nuove versioni principali in caso di modifiche importanti al prodotto o alle sue dipendenze.

Per i dettagli sulle nuove funzionalità disponibili in ogni versione, consulta. the section called "AWS Note sulla versione di Transform for mainframe"

Versioni preliminari Alpha

Ogni versione preliminare alfa è identificata da uno **`[Major].[Minor].0`**schema.

Le pre-release Alpha sono versioni frequenti e di breve durata, destinate e disponibili per una rapida iterazione durante i progetti di modernizzazione. Non esiste una cadenza di rilascio fissa per le nuove versioni pre-release Alpha e vengono rese disponibili man mano che vengono sviluppate e testate.

Per ulteriori informazioni su controllo delle versioni, aggiornamenti e supporto, consulta. Ciclo di vita dei componenti

> **Important**
>
> Le versioni preliminari di Alpha devono essere utilizzate solo durante la fase del progetto di modernizzazione e non per la produzione o i carichi di lavoro critici.

AWS Transform for mainframe Runtime

AWS Transform for mainframe offre un unico runtime per soddisfare le diverse fasi del percorso di modernizzazione e le esigenze operative. Questa pagina ne descrive le caratteristiche, i casi d'uso e come accedervi.

AWS Transform for mainframe Runtime

Con AWS Transform for mainframe Runtime puoi implementare la tua applicazione modernizzata autonomamente Account AWS , consentendoti di gestire la tua infrastruttura. Questa opzione offre sia la versione release che la versione alpha pre-release, offrendovi la flessibilità necessaria per

utilizzare tutti i componenti tecnici necessari per eseguire l'applicazione modernizzata nel modo desiderato. Puoi scegliere tra versioni stabili per ambienti di produzione o versioni alfa pre-release per scopi di test e sviluppo.

AWS Transform for mainframe Runtime viene distribuito e gestito dal cliente, offrendo un maggiore controllo sull'ambiente di runtime. Fornisce funzionalità di refactoring automatizzato ed è adatto per scenari di implementazione personalizzati.

Quando utilizzare

AWS Transform for mainframe Runtime è adatto per ambienti di test e produzione ed è particolarmente utile quando è richiesta una personalizzazione specifica dell'ambiente di runtime.

Come accedere

AWS Transform for mainframe Runtime è accessibile da AWS Transform for mainframe refactor Toolbox.

> **Note**
>
> L'accesso a AWS Transform for mainframe refactor Toolbox viene fornito nell'ambito del progetto AWS Transform for mainframe.

Implementazione

AWS Transform for mainframe Runtime è disponibile per la distribuzione su:

- Amazon EC2
- Amazon ECS su Amazon EC2
- Amazon EKS su Amazon EC2
- Amazon ECS gestito da AWS Fargate

La distribuzione su Amazon EC2 può essere eseguita direttamente nell'istanza o tramite un'applicazione containerizzata Docker, che è il modo preferito quando si utilizza Amazon ECS o Amazon EKS.

Per istruzioni dettagliate sulla distribuzione, consulta la documentazione di Configurazione Transform for mainframe di AWS Runtime.

AWS Note sulla versione di Transform for mainframe

Questa sezione contiene le note di rilascio di AWS Transform for mainframe Runtime and Modernization Tools dalla versione 3.5.0 in poi, la più recente, organizzate per numero di versione.

Per ogni versione, AWS Transform for mainframe Runtime è stato convalidato su versioni specifiche di strumenti (Tomcat, Postgres, Oracle, MQ, ecc.) e dipendenze (Spring, Angular, ecc.). L'utilizzo di altre versioni (ad esempio, l'aggiornamento a versioni più recenti) richiede test e convalide approfonditi da parte del cliente.

> **Note**
>
> Per le note di rilascio precedenti a questo documento, contatta AWS Transform for mainframe delivery services. Per informazioni sulle più recenti funzionalità di refactor di AWS Transform for mainframe, consulta AWS Transform for mainframe refactor releases.

Note di rilascio 5.75.0

Rilasciato il: 16 febbraio 2025

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines introduce miglioramenti significativi alle funzionalità della piattaforma, all'ottimizzazione delle prestazioni e alle tecnologie modernizzate. Alcune funzionalità e modifiche chiave includono:

- Miglioramenti completi dell'utilità JCL

  Maggiore compatibilità con i mainframe grazie a miglioramenti significativi su più utilità.

  - SORT/ICETOOL ora supporta operazioni OUTFIL avanzate, conversioni flessibili del formato di data (da giuliano a georgiano, Y4T, Y2P) e una migliore gestione dei campi a lunghezza variabile.
  - I miglioramenti apportati a DSNUTILB includono funzionalità migliorate del comando LOAD con opzioni POSITION e una migliore gestione delle colonne DECIMAL.
  - I miglioramenti di IDCAMS includono un solido supporto dei comandi ALTER e REPRO per i set di dati VSAM, la convalida completa dei file con codici di errore corretti e una migliore gestione del codice di ritorno tra le operazioni DELETE e PRINT.
- Gestione e prestazioni avanzate dei dati Blusam

I miglioramenti di Blusam ottimizzano le operazioni dei set di dati con un migliore riscaldamento selettivo per set di dati multischema, un supporto avanzato della cache Redis per file KSDS di grandi dimensioni e operazioni di chiusura più veloci del 60-88% grazie alla gestione ottimizzata degli indici.

- AS400 Prestazioni di posizionamento DAO migliorate

  La generazione ottimizzata di query SQL per le operazioni di posizionamento migliora significativamente le prestazioni del database. Il tempo di esecuzione delle query è stato ridotto di oltre il 99%, con l'elaborazione end-to-end in batch migliorata di 220 volte (da 11 minuti a 3 secondi) e l'accelerazione delle singole query di 7.161 volte (da 3.103 ms a 0,36 ms) sui carichi di lavoro di produzione.

Questa versione di Transform for mainframe Runtime è stata testata con il seguente stack AWS :

Componente	Versione testata
Java	Java 21
Livello di presentazione	Nodo JS 22.17.1
	Npm 10.9.0
	Angolare 20
Livello di servizio	Spring Boot 3.5.7
	Spring Core 6.2.12
	Sessione primaverile 3.5.2
	Spring Statemachine 4.0.0
Livello di persistenza	Motore PostgreSQL 15.10
	Oracle 21c
Report	Jasper 6
Server applicazioni	Apache Tomcat 10.1.40

AWS Transform per mainframe Runtime

ZoS

Miglioramenti

- COBOL
  - È stato aggiunto il supporto per i quaderni SQL con il livello 01 mancante
  - Supporto migliorato per l'istruzione MOVE da LOW VALUES a dimensione variabile
  - Supporto migliorato per l'istruzione MOVE to COMP-X
  - Supporto migliorato per l'istruzione DISPLAY con clausola WITH CONVERSION
  - Supporto migliorato per l'istruzione SORT per file sequenziali di righe con lunghezza variabile e dipendente dalla clausola
  - Supporto migliorato per l'istruzione INSPECT con più campi di conteggio
  - Supporto migliorato per l'istruzione INSPECT for TAYLLING con AFTER e BEFORE
  - Supporto migliorato per il funzionamento delle sottostringhe per il tipo di dati GRAPHIC
  - Supporto migliorato per i valori HIGH e LOW per evitare l'overflow del buffer causato da dimensioni del buffer non corrispondenti tra i record di controllo COBOL e i parametri del programma associato
  - È stato aggiunto il supporto per l'utilità di sistema IEBDG
- JCL - ORDINA
  - Supporto migliorato per OVERLAY e conversione della data dal formato giuliano a quello georgiano
  - Supporto migliorato per il formato Y4T nelle istruzioni OUTFIL
  - Supporto migliorato per la frase DEBUG ABEND alla fine della scheda
  - Supporto migliorato per OUTFIL INCLUDE (), REMOVECC, NODETAIL, TRAILER1
  - Supporto migliorato per la parola chiave OUTFIL BUILD
  - Supporto migliorato per il formato dell'anno Y2P (Year to Packed)
  - Supporto migliorato per le parole chiave TIMENS e TIME
  - Supporto migliorato per la parola chiave CENTWIN
  - Supporto migliorato per gli spazi vuoti e la parola chiave COUNT
  - Supporto migliorato per Limita il numero e i caratteri del trailer e per l'avanzamento dei caratteri solo su record selezionati definiti dalla condizione INCLUDE/OMIT

  - Supporto migliorato per i tipi di dati alfanumerici con prefisso di dimensione durante l'elaborazione di campi alfanumerici a lunghezza variabile con prefissi di dimensione
  - Supporto migliorato per la conversione del formato TRAILER1 TO= con LENGTH in TOTAL
  - Supporto migliorato per JFY =(SHIFT=LEFT/RIGHT), TRAIL, LEAD quando il record contiene uno o più campi
  - È stato aggiunto il supporto per la sintassi BUILD (...) oltre alla sintassi BUILD =(...)
  - È stato aggiunto il supporto per il carattere di continuazione del trattino
  - È stato aggiunto il supporto per il confronto tra byte e numeri interi della BI
- JCL - DSNUTILB
  - Supporto migliorato per il comando LOAD con l'opzione POSITION (start:end)
  - Supporto migliorato per il comando LOAD con colonne di tipo DECIMAL (senza EXTERNAL)
  - Supporto migliorato per il comando LOAD durante la gestione di RDW nel contesto SORT
  - Supporto migliorato per l'opzione NULLIF
  - Migliore costruzione del percorso del file di output durante l'operazione COPY
- JCL - IDCAMS
  - Supporto migliorato per l'istruzione ALTER per gestire file non VSAM
  - Supporto migliorato per l'istruzione REPRO per la gestione del parametro REPLACE per il set di dati KSDS
  - Supporto migliorato per il comando REPRO per gestire la convalida file/dataset dell'esistenza di input e output che restituisce RC=12 con il codice di errore U4020 in caso di file mancanti
  - Supporto migliorato per i codici di ritorno per il comando DELETE
  - Supporto migliorato per SPANNED/NONSPANNED e RECOVERY
  - Supporto migliorato per GDG Current Generation quando IDCAMS elimina GDG preceduto dalla nuova generazione
  - Supporto migliorato per il codice di ritorno nel caso del comando PRINT
- JCL - ICETOOL
  - Supporto migliorato per i pattern di SORT, SPLICE, SELECT e OCCUR per accettare qualsiasi ordine dei relativi parametri
  - Supporto migliorato per l'analisi flessibile degli ordini degli operandi e il codice di restituzione
  - Supporto migliorato per l' COPY/SORT assenza del parametro TO quando non è definito alcun file di output nel comando OUTFIL

- JCL - Varie
  - Supporto migliorato per il simbolo del dollaro nell'esecuzione di PROC
  - Supporto migliorato per set di dati temporanei nell'elaborazione JCL
  - Supporto migliorato per l'istruzione DD per gestire i commenti in linea tra parentesi
  - Supporto migliorato per la sostituzione delle variabili JCL
  - Supporto migliorato per i programmi IKJ e i parametri PARM
  - È stato aggiunto il supporto per i parametri JCL DD, inclusi DD * (asterisco) e DD
- Blusam
  - Processo di riscaldamento selettivo migliorato per il caricamento della cache dei set di dati multischema rimuovendo la distinzione tra maiuscole e minuscole in BlueSam.FileList
  - Processo di riscaldamento migliorato, in cui il flag di riscaldamento viene impostato due volte durante le operazioni combinate di caricamento dei dati e di esecuzione del programma, evitando operazioni di riscaldamento duplicate e garantendo un comportamento coerente in diversi scenari di esecuzione
  - Supporto migliorato per file KSDS di grandi dimensioni contenenti un solo record e tale record viene eliminato tramite l'operazione REWRITE
  - Supporto migliorato per Redis Cache quando i set di dati vengono eliminati utilizzando Blusam delete
  - Supporto migliorato per il parametro Redis TTL nel metodo put
  - Supporto migliorato per Redis e KSDS di grandi dimensioni durante le operazioni di scrittura di set di dati ad alto volume
  - Supporto migliorato per i valori DataSet Type nella creazione del set di dati
  - Prestazioni migliorate del 60-88% nelle operazioni di chiusura per set di dati KSDS di grandi dimensioni, grazie alla possibilità di evitare il recupero di carichi utili dell'indice di grandi dimensioni quando è necessario solo il controllo dello stato degli aggiornamenti
  - BAC - Endpoint UpdateRecord migliorato per utilizzare la chiave primaria se non viene passato alcun id
- SQL
  - Supporto migliorato per la clausola «PARTITION BY» per accettare più argomenti
  - Supporto migliorato per l'istruzione EXTRACT e i parametri TIME con HOURS/MINUTES/SECONDS
- IMS

  - Ha implementato la generazione di chiavi concat basate su Epoch Timestamp per segmenti HSAM senza campi di sequenza, garantendo che ogni segmento abbia una chiave di navigazione valida
  - Supporto migliorato per le chiamate GNP con il comando D
  - Supporto migliorato per l'istruzione ISRT con SSA Parent qualificato all'inserimento dei record rimuovendo la logica di ricerca della posizione corrente del database che ha la precedenza su qualified SSAs
  - Supporto migliorato per l'istruzione ISRT per consentire l'inserimento del comando SSA con D e tutto ciò che SSAs segue nel DB per le tabelle logiche
  - Migliore inizializzazione dello stato del PCB
  - Supporto migliorato per la registrazione I/O PCB
  - Supporto migliorato per le chiamate PLITDLI con segmenti definiti in DBD senza un genitore
  - Supporto migliorato per il comando GNP sul segmento logico di terzo livello
  - Supporto migliorato per l'associazione di pagine logiche in caso di associazioni dirette MID vuote
  - Supporto migliorato per il formato PFK misto e implementazione della funzione di controllo NEXTLP
  - Supporto migliorato per l'inserimento di segmenti accoppiati con genitori logici per risolvere i nomi dei database principali logici e implementare l'algoritmo di selezione cumulativa dei campi per la generazione accurata delle condizioni JOIN nelle viste logiche
- CICS
  - Supporto migliorato per il comando RETURN per gestire i campi di gruppo
  - Supporto migliorato per il comando ENTER con TRACEID e TRACENUM
  - Supporto migliorato per il comando ENTER con campi di gruppo
  - Supporto migliorato per il parametro WRITE OPERATOR ACTION e i codici di azione numerici (2=IMMEDIATE, 3=EVENTUAL, 11=CRITICAL)
  - Supporto migliorato per il comando WRITE OPERATOR TEXT per i campi con pedice
  - Supporto migliorato per il comando SOAP
  - Supporto migliorato per il comando GET CONTAINER con l'opzione SET
  - Supporto migliorato per ADDRESS EIB nei programmi NOLINKAGE COBOL
  - Supporto migliorato per il comando CICS INQUIRE FILE con l'opzione ENABLEDSTATUS
  - Migliore propagazione del canale corrente CICS
  - Supporto migliorato per SYNCPOINT con cursori SQL WITH HOLD

- Supporto migliorato per il comando DEQ quando non esiste alcuna coda
- Supporto migliorato per il comando RECEIVE con il parametro FLENGTH contenente un riferimento variabile
- Supporto migliorato per il parametro XCTL INPUTMSG
- Aggiunto il supporto per il comando FETCH ANY
- Aggiunto un ordine flessibile per CICS MAP e MAPSET

- PL/1
  - Aggiunto il supporto per la trasformazione dei servizi richiamabili Language Environment (LE) per i programmi PL/I
  - È stato aggiunto il supporto per le funzioni integrate DAYS e DAYSTODATE
  - Aggiunto il supporto per l'argomento LABEL per il comando ABEND CICS
- MQ
  - Gestore di code migliorato per disconnettere le code solo al termine dell'unità di esecuzione
  - Aggiunto il supporto per le proprietà ReplyTo Q e ReplyTo QMgr nell'operazione MQ JSM Put

AS400

Miglioramenti

- RPG
  - Supporto migliorato delle specifiche e end-of-file dei record delle estensioni
  - Supporto migliorato della sottoprocedura che restituisce il tipo booleano
  - Supporto migliorato della sottoprocedura che restituisce il tipo di puntatore
  - Supporto migliorato per l'extender R in EVAL
  - Supporto migliorato del funzionamento MOVEA per gestire l'extender P
  - È stato aggiunto il supporto per la funzione integrata %REALLOC per riallocare lo storage degli heap a una nuova lunghezza, preservando i dati esistenti
  - Supporto migliorato per *ZEROS figurativi utilizzati come valore di ritorno
  - Supporto migliorato per il formato di data*LONGJUL
- COBOL
  - Supporto migliorato per la frase USING quando il parametro in è una sottostringa
  - Supporto migliorato per l'istruzione «INSPECT» con valori letterali esadecimali

  - Supporto migliorato dell'area degli indicatori in modalità separata (INDARA) nelle operazioni sui file di visualizzazione. I/O
- CL
  - Progettazione generale migliorata della gestione dei file per gestire il parametro OPNID del comando RCVF
  - Supporto migliorato dei parametri per il comando SBMJOB per gestire valori vuoti
  - Supporto migliorato per il comando SBMJOB per il parametro RQSDTA senza CMD
  - Supporto migliorato del comando CPYF per copiare da una tabella a un file flat quando mancano gli argomenti CRTFILE (*YES) e MBROPT
  - Supporto migliorato per il comando SAVOBJ per gestire gli spazi nel parametro SAVF
  - Supporto migliorato per il comando CLLE per il confronto con stringhe vuote
  - Supporto migliorato di DLTF per la gestione del parametro posizionale FILE
  - Supporto migliorato per il comando CHKOBJ per la gestione dei parametri
  - Supporto migliorato per il comando OVRDBF su LF con più PFILE
  - Supporto migliorato per il comando OVRDBF per gestire la risoluzione degli elenchi di librerie
  - Supporto migliorato, comando di analisi dei parametri posizionali QCMDEXC con chiamata a OVRDBF e DLTOVR
  - Supporto migliorato per il comando startpos/length CHGDTAARA e i parametri passati come variabili
  - Migliore supporto per messaggi di monitoraggio generici CPF9999 e CPF0000 per catturare qualsiasi tipo di messaggio di errore
  - Supporto migliorato per il comando RMVLIBLE senza parametri
  - Supporto migliorato del parametro MSGQ sul comando RCVMSG
  - È stato aggiunto il supporto per lo schermo e il comando di DSPMSG per visualizzare i messaggi inviati a una coda di workstation.
  - Implementata la risoluzione dei nomi delle colonne di query JPA in OPNQRYF
  - È stato aggiunto il supporto per SNDBRKMSG per l'invio di un messaggio immediato a una workstation
  - Supporto migliorato per *COMP negli script groovy
  - Supporto migliorato per il comando DCLF sui file di visualizzazione con campo a più condizioni
  - Supporto migliorato di CHGVAR per gestire la conversione da numerico a alfanumerico

  - Supporto migliorato per il taglio del valore su cpytoimpf

  - Supporto migliorato del comando FTP per gestire l'istruzione APPEND
- Accesso al database
  - Meccanismo di cache DAO migliorato per garantire letture sequenziali affidabili quando i record vengono modificati contemporaneamente, mantenendo prestazioni costanti.
  - Prestazioni migliorate dei componenti di controllo dell'esistenza di tabelle e partizioni
  - Supporto migliorato per l'override dei file di database di più entità
  - Creazione e registrazione DAO migliorate sulla configurazione multithread
  - Supporto migliorato del flag EOF in DAO
  - Migliore inizializzazione del DAO multiformato
  - Supporto delle partizioni migliorato per DAO dinamici
  - Prestazioni migliorate sulle operazioni DAO ottimizzando le query SQL e sfruttando la sintassi di confronto delle righe di PostgreSQL, riducendo il tempo di esecuzione delle query del 99,5% (da 11 minuti a 3 secondi su tabelle di grandi dimensioni).
  - Supporto migliorato del membro sulla libreria QTEMP
  - Supporto migliorato dell'elenco delle librerie con area dati
  - Supporto della cache migliorato per le operazioni di input con entità di file flat
- Schermo
  - Supporto migliorato delle operazioni READ sui record dei terminali
  - Supporto migliorato per i campi ambigui all'interno dei record dei sottofile
  - Supporto migliorato per la parola chiave CHANGE per il rilevamento delle modifiche dei campi
  - Comportamento ERRMSG migliorato quando si lavora con indicatori compositi
  - Supporto migliorato per la parola chiave SFLRNA per creare sottofile senza record attivi
  - Maggiore robustezza della comunicazione tra front-end e back-end durante i messaggi di interruzione
  - Migliorata la visualizzazione dei campi sovrapposti sul terminale
  - Supporto migliorato per le parole chiave DSPF ASSUME e KEEP quando le dimensioni dello schermo cambiano
  - Supporto migliorato per la messa a fuoco sullo schermo senza input modificabili per consentire alla schermata di aiuto (F1) di cercare la descrizione corretta relativa al record attivo
  - Aggiunto il supporto alla generazione delle parole chiave ASSUME e KEEP per COBOL
  - Supporto migliorato per CHGINPDFT (LC) sul frontend

    - Supporto migliorato per la parola chiave USRRSTDSP
- Stampante
    - È stato aggiunto il supporto per le parole chiave di controllo della stampante Advanced Function Presentation Data Stream (AFPDS) ENDPAGE, LINE, PAGSEG, OVERLAY, FNTCHRSET
    - Layout del report di testo migliorato
    - Dipendenza Jasper aggiornata alla versione 7
    - Supporto migliorato per i campi PRTF che sostituiscono quelli nei file PF
    - È stato aggiunto il supporto per l'ID del carattere (2308/2309)
- Processo
    - È stato aggiunto il supporto per la configurazione predefinita del formato della data di lavoro, migliorando la flessibilità di gestione della data in base alle proprietà YML system.date.format e system.date.separator (new)
    - Aggiunto il supporto completo per il filtraggio dei contenuti dei messaggi all'API Job Queue con funzionalità di filtraggio multicampo
- Spazio utente
    - Migliore rimozione degli oggetti
    - Aggiunto il supporto per l'indicatore di errore nelle funzioni User Space
    - Prestazioni migliorate sul supporto Redis
- Misc (Vari)
    - Conversione in testo semplice migliorata di numeri con esponenti
    - Migliore risoluzione dell'array di byte a partire da un riferimento a un intervallo di array indicizzato
    - Supporto migliorato della sessione attiva su Redis
    - Supporto migliorato per i dati passati a un'API Data Queue.
    - Migliore persistenza interattiva dei lavori

AWS Transform per motori di trasformazione mainframe

ZoS

Miglioramenti

- COBOL
    - È stato aggiunto il supporto per l'istruzione PROGRAM COLLATING SEQUENCE

- È stato aggiunto il supporto per il simbolo della sterlina (£,\ x00A3) nelle clausole PIC [TURE]
- Supporto migliorato per la determinazione delle dimensioni dell'area di comunicazione (COMMAREA) per scenari non dichiarati
- Supporto migliorato per l'istruzione INSPECT con REPLACING phrase e SPACES
- Supporto migliorato per la definizione dei campi per gestire la parola chiave OCCURS riservata come nome valido
- Supporto migliorato per l'istruzione UNSTRING quando gli elementi di destinazione sono separati da una virgola
- Supporto migliorato per la clausola COPY REPLACING
- Supporto migliorato per l'istruzione CALL con le clausole CALL e GIVING INTO
- Supporto migliorato per INSERT e UPDATE con clausola RETURNING
- Supporto migliorato per il campo implicit/explicit SQL-INIT-FLAG
- Supporto migliorato per la variabile LEVEL 88 con VALUE ZEROS nell'elemento di gruppo
- Supporto migliorato per le macchine a stati e l'evento di continuazione per il ciclo di ricerca finale
- Supporto migliorato per le macchine Perform Section State
- Supporto migliorato per l'istruzione JSON GENERATE con funzionalità RENAME per utilizzare percorsi di campo COBOL completi anziché semplici nomi di campo

- JCL
  - Supporto migliorato JCL PROCs contenente parametri simbolici con il nome «SET»
- CICS
  - Aggiunto il supporto per l'opzione di traduzione NOLINKAGE per impedire la riscrittura della sezione di collegamento e della divisione delle procedure
  - Supporto migliorato per il collegamento DFHCOMMAREA
  - Aggiunto il supporto delle opzioni DELAY FOR MINUTES e HOURS
- SQL
  - È stato aggiunto il supporto per i quaderni SQL con 01 LEVel mancante
  - Supporto migliorato per l'istruzione EXEC SQL WHENEVER NOT FOUND
  - Supporto migliorato per l'istruzione SQL USAGE IS con TYPE IS BLOB
  - Analisi migliorata delle istruzioni del Data Definition Language con una migliore gestione delle parole chiave come identificatori
  - Supporto migliorato per Type ROWID

- Supporto migliorato per VARCHAR con parametri come (",1)

AS400

Miglioramenti

- RPG
  - Supporto migliorato per il funzionamento MOVEA con array di indicatori *IN
  - Supporto migliorato per le operazioni MOVE per campi autonomi con lunghezza diversa
  - Supporto migliorato per le operazioni MOVE con struttura dati di dimensioni diverse
  - Supporto migliorato per il funzionamento e gli indicatori MOVE
  - Supporto migliorato per le parole chiave Data Structure OVERLAY e ALIGN
  - Migliore risoluzione del campo in tutta la sottoprocedura
  - Supporto migliorato delle funzioni integrate %INT, %INTH e %STR
  - Supporto migliorato del funzionamento GOTO per rimuovere il codice irraggiungibile dopo l'istruzione RETURN
  - Supporto migliorato dell'istruzione ENDSR nelle strutture di controllo annidate
  - Supporto migliorato della clausola BY VALUE su un parametro di sottoprocedura per impedire la modifica del valore
  - Supporto migliorato della parola chiave LIKEDS per gestire il caso di copia delle proprietà e dei campi di una struttura di dati utilizzando la parola chiave ALIGN
  - Supporto migliorato per l'inizializzazione di array fissi alfanumerici
  - Supporto migliorato per %SCAN per l'utilizzo dell'indicizzazione basata su 0
  - Supporto migliorato per il funzionamento RESET con la subroutine *INZSR
  - Supporto migliorato per la parola chiave ON-ERROR
  - Supporto migliorato per il funzionamento LOOKUP con indicatori HI/LO/EQ
  - Supporto migliorato del parametro di procedura definito come costante con una parola chiave LIKE
  - È stato aggiunto il supporto per la parola chiave dell'area dati *NAMVAR
  - È stato aggiunto il supporto per il funzionamento EVAL-CORR su formato fisso
  - Supporto migliorato per l'assegnazione di valori esadecimali (MOVE ed EVAL)
  - Supporto migliorato della clausola ON-EXCEPTION per la generazione di API per la coda di messaggi

  - Supporto migliorato per QCLRDTAQ con gruppi di parametri aggiuntivi
  - Supporto migliorato per l'istruzione Return estraendo ogni chiamata annidata e sostituendola con la relativa variabile return
- DDS
  - Migliore gestione dei campi di riferimento per evitare riferimenti circolari
  - Supporto migliorato della parola chiave JREF per i file logici JOIN
  - Supporto migliorato della parola chiave COMP per gestire il riferimento a un altro campo
  - È stato aggiunto il supporto per i tipi di campo BLOB con dimensioni specificate
  - È stato aggiunto il supporto per ALTER TABLE ADD COLUMN
- CL
  - È stato aggiunto il supporto del tipo compresso e suddiviso in zone senza segno per supportare il tipo numerico definito con una parola chiave di modifica specifica (EDTCDE/EDTWRD)
- PRTF
  - Generazione di modelli di report migliorata per migliorare i calcoli della banda e dell'altezza della pagina
  - Aggiunto il supporto per la generazione di modelli per le parole chiave della stampante AFPDS (ENDPAGE, LINE, PAGSEG, OVERLAY, FNTCHRSET) e la parola chiave POSITION
- COBOL
  - Supporto migliorato per le utilità di coda dati IBM QCLRDTAQ, QRCVDTAQ e QSNDDTAQ
  - Supporto migliorato dell'istruzione COPY DDS per generare il suffisso RECORD solo nella sezione File
  - Supporto migliorato del file descritto dal programma con più record
  - Migliore generazione di record DDS tramite un'istruzione COPY DDS
  - Risoluzione dei campi migliorata per gestire nomi di campo duplicati quando sono presenti riferimenti a parole chiave LIKE
  - Risoluzione dei campi migliorata quando una struttura utilizza la parola chiave LIKE per fare riferimento a un record DDS; i riferimenti ai campi ora risolvono correttamente la definizione DDS originale per impostazione predefinita.
  - È stato migliorato il supporto della clausola COPY DDS nella descrizione del file per generare l'entità record DDS nel contesto del programma
  - È stato aggiunto il supporto per i valori letterali esadecimali nell'istruzione INSPECT
  - Supporto migliorato della clausola COPY per DSPF

  - Supporto migliorato per l'istruzione MOVE con valori letterali esadecimali NX
  - È stato aggiunto il supporto RRN per il funzionamento dell'output della workstation COBOL
  - Supporto migliorato per le chiamate dinamiche quando si utilizza la sottostringa
  - Supporto migliorato per DO HIVAL per evitare loop infiniti
  - Supporto migliorato di condizioni binarie complesse con una combinazione di operatori OR e AND
  - Supporto migliorato degli indicatori con il funzionamento REWRITE SUBFILE
- DAO
  - Generazione migliorata di numeri interi di grandi dimensioni
  - È stato migliorato il supporto dei campi di condizione definiti in un file logico JOIN
  - Supporto migliorato per l'etichettatura dei nomi di colonne lunghe in SQL

AWS Transform for mainframe Transformation Engine e Runtime for GS21

Miglioramenti

- Record gestiti appartenenti a set multi-membro per l'utilità JXHDBCLR

Note di rilascio 5.1.0

Rilasciato il: 18 novembre 2025

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines introduce miglioramenti significativi alle funzionalità della piattaforma, all'ottimizzazione delle prestazioni e alle tecnologie modernizzate. Alcune funzionalità e modifiche chiave includono:

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

- AWS Transform for mainframe Transformation Engines & Runtime for GS21

  È stato aggiunto il supporto del mainframe proprietario di Fujitsu GS21 nell'ambito della partnership MAJI (Modernization Acceleration Joint Initiative). Le caratteristiche principali includono diversi nuovi linguaggi come PSAM, ADL, ACSGEN o JCL XSP, la modernizzazione del database di rete NDB, l'implementazione di ecosistemi come AIM, OFM e ACSAPI e il supporto della codifica JEF. GS21

- Aggiornamento della versione Java

  Aggiornamento da Java 17 a Java 21, 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

Questa versione di AWS Transform for mainframe Runtime è stata testata con il seguente stack:

Componente	Versione testata
Java	Java 21
Livello di presentazione	Nodo JS 22.17.1
	Npm 10.9.0
	Angolare 20
Livello di servizio	Spring Boot 3.5.7
	Spring Core 6.2.12
	Sessione primaverile 3.5.2
	Spring Statemachine 4.0.0
Livello di persistenza	Motore PostgreSQL 15.10
	Oracle 21c
Report	Jasper 6
Server applicazioni	Apache Tomcat 10.1.40

AWS Transform per mainframe Runtime

ZoS

Miglioramenti

- COBOL

  - Aggiunto supporto per le routine di utilità CEELOCT, CEESECS, CEEDATE
  - Supporto migliorato per l'istruzione JSON GENERATE e l'opzione COUNT
  - Supporto migliorato per la clausola BY VALUE nell'istruzione PROCEDURE DIVISION
  - Supporto migliorato per l'istruzione COMPUTE per gestire la rappresentazione esponenziale dei letterali numerici
  - Supporto migliorato per il comando CICS START per le opzioni TRANS e SYSID
  - Supporto migliorato per l'istruzione MOVE per gestire l'overflow numerico letterale
  - Supporto migliorato per il ridimensionamento delle divisioni per tipi a precisione fissa
  - Supporto migliorato per l'istruzione CALL con clausola USING
- JCL - ORDINA
  - Aggiunto il supporto per il formato DATE «4MD/»
  - Aggiunto il supporto per la parola chiave RESTART
  - Aggiunto il supporto per le parole chiave DATENS e TIMENS
  - Aggiunto il supporto per il formato Y2S
  - Supporto migliorato per l'istruzione OUTFIL con più file
  - Supporto migliorato per i campi REMARKS
  - Supporto migliorato per l'istruzione OUTREC e l'associazione SYMNAMES
  - Supporto migliorato per i controlli numerici e alfanumerici per l'istruzione INCLUDE
- JCL - DSNUTILB
  - Supporto migliorato per COPYDDN quando il file GDG viene utilizzato come parametro
- JCL - IDCAMS
  - Supporto migliorato per la definizione di set di dati con virgolette singole
- JCL - Misc
  - Aggiunto il supporto per gestire il passaggio di PARM vuoti al programma tramite JCL
  - Aggiunto il supporto per l'utilità INZUTILB
  - Aggiunto il supporto per il numero di generazione parametrizzato nell'istruzione GDG DD
  - Lettura migliorata dei file SYSIN con dimensioni record
- Blusam
  - Cambia il valore predefinito per la proprietà openwarmup da true a false
  - Aggiunta una nuova proprietà YML useBatchIn MemoryAtomic per abilitare la in-memory-atomic modalità per le operazioni di lettura in batch di Redis

  - Supporto migliorato per la riscrittura dei set di dati indicizzati quando viene rilevata una chiave non valida
  - Supporto migliorato per la creazione di tabelle interne KSDS di grandi dimensioni con tipo di dati OID
  - Supporto migliorato per i dati KSDS di grandi dimensioni letti durante il recupero della pagina previous/next
- SQL
  - Supporto migliorato per le variabili host nell'istruzione SQL CONNECT
  - Supporto migliorato per la conversione dei dati nelle query SQL
  - Supporto migliorato per le query parametrizzate con identificatori tra virgolette doppie
- IMS
  - Supporto migliorato per NULLVAL nell'istruzione XDFLD
  - È stato aggiunto il supporto per l'istruzione SEG e l'istruzione END multipla
- CICS
  - Supporto migliorato per il comando RETURN per gestire l'opzione INPUTMSG

AS400

Miglioramenti

- RPG
  - Aggiunto il supporto per più tabelle in LOOKUP for CTDATA
  - Aggiunto il supporto per il parametro*OMIT nella gestione del codice di feedback dell'API CEELOCT
  - Aggiunto il supporto per *LOVAL assegnato ai tipi di campo data, ora e timestamp
  - Aggiunto il supporto per convertire l'espressione in timestamp ISO0
  - Calcolo della dimensione dei byte migliorato per l'array indicizzato
  - Supporto migliorato del funzionamento MOVEA tra tipi di array eterogenei suddivisi in zone e compressi
  - Supporto migliorato del funzionamento MOVEA dall'array definito in DS con OCCURS
  - Supporto migliorato di MOVEA con array basato su una struttura dati esterna
  - Accesso agli indicatori dinamici migliorato utilizzando una variabile con MOVEA
  - Supporto migliorato per l'istruzione MOVE dalla struttura dei dati al campo autonomo

  - Supporto migliorato per %DEC integrato annidato (%SUBST ())
  - Supporto migliorato per la parola chiave EXTFILE per utilizzare l'elenco delle librerie solo quando la libreria non è specificata
  - Supporto migliorato per EDITCODE con le opzioni A, J, P, K e CHECK (RB)
  - È stato aggiunto il supporto della parola chiave *JOBRUN per impostare i separatori per il tipo Data e ora
- CL
  - Aggiunto il supporto per il comando QRCVDTAQ
  - Aggiunto il supporto per il comando RNMOBJ
  - Supporto migliorato per il comando CPYFTOIMPF che include valori decimali inferiori a 1 e valori vuoti
  - Supporto migliorato per il comando CPYTOIMPF per preservare l'ordine di inserimento e generare spazi singoli per stringhe vuote in colonne non annullabili
  - Supporto migliorato per il comando OPNQRYF con l'opzione «QRYSLT» e colonne DB diverse dai nomi dei campi delle entità Java
  - Supporto migliorato per il comando OPNQRYF quando i nomi dei campi di entità non corrispondono ai nomi delle colonne del DB
  - Supporto migliorato per il comando OPNQRYF con parametri KEYFLD contenenti valori MAPFLD
  - Supporto migliorato per il comando SNDPGMMSG con i parametri MSGTYPE (*STATUS) e TOPGMQ (*EXT)
  - Supporto migliorato per il comando CPYF e l'opzione MBROPT quando la partizione non esiste e CRTF (*YES)
  - Supporto migliorato per il comando CPYF con convalida numerica dei dati e gestione degli errori mediante l'aggiunta di controlli di pre-conversione e fallback a valore zero
  - Supporto migliorato per il comando CPYF con parametro INCREL
  - Supporto migliorato per il comando CPYF per gestire l'inserimento manuale di righe per gestire i campi numerici NULL
  - Supporto migliorato per il comando SBMOBJ per recuperare il nome utente nel contesto in cui viene definito e altrimenti con il valore predefinito JOBD
  - Supporto migliorato per il comando RMVMSG per i parametri MSGQ e MSGKEY
  - Supporto migliorato per il formato di output del comando DSPDBR

- Supporto migliorato del comando CHGDTAARA per gestire l'area dati locale
- Migliore precisione dell'output dei comandi DSPDBR implementando la corretta gestione delle maiuscole, i valori predefiniti e la spaziatura interna, garantendo al contempo una numerazione dei record e una visualizzazione coerenti dei nomi dei membri
- Supporto migliorato per il comando RTVJOBA per la gestione dell'attributo USRLIBL
- Supporto migliorato per il comando MOVOBJ per gestire file flat
- Supporto migliorato per il comando RMVM per gestire File come parametro RangeReference
- Supporto PgmAlcObj e PgmDlcObj comandi migliorati per ignorare le operazioni di blocco per i file flat
- Supporto migliorato di OVRDBF con file flat e *LIBL
- Supporto migliorato di RTVOBJD per restituire il nome LIB con il file su disco
- È stato aggiunto il supporto per il messaggio CPF999 che cattura tutti i messaggi non gestiti
- Migliore recupero delle visualizzazioni per RTVMBRD senza WhereConditions

- Accesso al database
  - Funzionalità avanzate di sovrascrittura dei file per supportare i file JOIN
  - Regole di precedenza avanzate degli AND/OR operatori durante l'elaborazione di più condizioni Select/Omit
  - Sincronizzazione simultanea della cache migliorata utilizzando il controllo delle versioni dei record basato su timestamp
  - Supporto migliorato per OVRDBF con operazione di scrittura in batch
  - Meccanismo di blocco dei record del database migliorato sull'accesso multithread
  - Migliore inizializzazione della cache durante l'operazione READ
  - Supporto migliorato per l'operazione DELETE in Commit Control
  - Supporto migliorato del file primario con file logici multiformato
  - Aggiunto il supporto di LIBL per DataArea e CURLIB in tutti i job
- Schermo
  - È stato migliorato l'escape delle virgolette singole per il valore String
  - Migliore pulizia dei componenti Web durante la gestione delle operazioni di scrittura senza sovrapposizione, in particolare per i componenti contrassegnati con la parola chiave keep
  - Supporto migliorato per la parola chiave SLNO
  - Supporto migliorato per Terminal ID

  - Supporto migliorato per numeri di grandi dimensioni in un campo di immissione
  - Posizionamento del cursore migliorato durante il caricamento delle tabelle nel front-end ordinando i file di visualizzazione colonna per righe e in ogni riga per colonne
  - È stato aggiunto il supporto per la parola chiave PUTTEAIN (Put-Retain) per i file di visualizzazione
  - Supporto migliorato di Terminal ID in una sessione online
- Stampante
  - Supporto migliorato per la generazione di numeri di pagina
  - È stato aggiunto il supporto per la gestione dell'overflow nei report
  - È stato aggiunto il supporto per la parola chiave INFDS per i file della stampante
- Processo
  - Migliore pulizia delle risorse lavorative al termine di una sessione online
  - Registrazione dei lavori Quartz migliorata per includere nome utente e JobNumber formattato nelle proprietà del lavoro
  - JobHistory Creazione migliorata per utilizzare dinamicamente NetName per sessioni interattive
  - Supporto migliorato per l'invio dei lavori e la convalida del formato della data
- Misc (Vari)
  - Aggiunto il supporto per la funzione di ordinamento EBCDIC CP1047 per Postgres
  - Spazio utente: parametri del codice di errore migliorati per il formato Errc0100 per lo spazio utente
  - Jedis è stato aggiornato alla versione 6.0.0 per migliorare la persistenza della sessione in Redis e ha esteso il supporto per il tracciamento delle sessioni su più endpoint.
  - Supporto migliorato per il programma composto quando il formato ID del programma contiene caratteri «/»
  - Migliore gestione dei campi DDS definiti come parametri del programma
  - Supporto migliorato del ErrorCode parametro per il comando QMHSNDPM
  - Supporto migliorato per la conversione della data da ISO a EUR

AWS Transform per motori di trasformazione mainframe

ZoS

Miglioramenti

- COBOL
  - Supporto migliorato per la continuazione di riga e la stringa multilinea
  - Supporto migliorato per l'istruzione INITIALIZE con più campi
  - Supporto migliorato per i quaderni contenenti l'istruzione Occurrents nelle dipendenze
  - Migliore generazione di formati di file rimuovendo i separatori di virgole nei valori numerici e aggiungendo il supporto di tipo binario per i campi COMP
  - Supporto migliorato per l'istruzione CALL utilizzando l'opzione BY VALUE ZERO
  - Supporto migliorato per l'istruzione JSON GENERATE per gestire la frase NAME
- CICS
  - Supporto migliorato per il comando LINK con l'opzione SYNCONRETURN
  - Supporto migliorato per il comando GET CONTAINER con l'opzione NODATA
  - Supporto migliorato per il comando WRITE OPERATOR TEXT per l'opzione ROUTECODES e NUMROUTES
  - Aggiunto il supporto per il comando INQUIRE TSPOOL
- IMS
  - Supporto migliorato per SLASHSX e DDATA nella trasformazione dei file DBD
  - Supporto migliorato per la parola chiave TITLE after PRINT nella mappa MFS
  - Aggiunto il supporto per le istruzioni COPY ed EQU, supporto per i file MFS
- SQL
  - Accetta la parola chiave FLOOR come nome della colonna
  - Supporto migliorato per i campi TIME-ZONE
  - Supporto migliorato per le sottoquery if condition e consective all'interno delle clausole FROM e JOIN

AS400

Miglioramenti

- RPG
  - Supporto migliorato del funzionamento DO
  - È stato aggiunto il supporto per gli indicatori di risultato durante il MOVE/MOVEL funzionamento
  - Supporto migliorato per il funzionamento MOVE per il campo array autonomo

- Supporto migliorato per il funzionamento MOVEA durante lo spostamento di valori dagli array agli indicatori
- Supporto migliorato del funzionamento Z-ADD con il campo dei risultati come array.
- Supporto migliorato per la parola chiave*Start per funzioni di accesso ai dati come SETLL
- Supporto migliorato per le procedure di chiamata con parametri passati da Value definiti in file esterni
- Supporto migliorato per l' sizePrefixedAlphanumeric utilizzo del figurativo *ALL
- Supporto migliorato per la funzione EXTIND
- Aggiunto il supporto per GOTOs l'istruzione nelle subroutine in cui le corrispondenti si TAGs trovano nella subroutine principale
- Supporto migliorato della parola chiave NOT in condizioni IFs/LOOPs
- Migliore ridenominazione dei campi tramite specifiche di input
- Migliore tracciamento dei «risultati scale/precision intermedi» per le moltiplicazioni
- Condizioni dell'indicatore delle specifiche di output implementate
- Supporto migliorato del funzionamento LOOKUP
- Supporto migliorato dei puntatori tra i programmi
- Supporto migliorato per %CHAR per gli input numerici
- Supporto migliorato della funzione incorporata %dec utilizzata con un solo argomento
- Migliore inizializzazione dei campi della struttura dei dati
- Migliore gestione del tipo di destinazione con numeri interi binari durante l'elaborazione del valore letterale *ALL
- Gestione migliorata *LOVAL con i tipi Packed, Float e Double

- CL
  - Supporto migliorato per l'istruzione DCL per gestire le occorrenze duplicate
  - Supporto migliorato per SUBROUTINE e istruzioni OTHERLY
- DDS
  - Rilevamento migliorato dei campi chiave per i campi di tipo compresso
  - Supporto migliorato della parola chiave LIKE nei file fisici
- DSPF
  - Supporto migliorato per la parola chiave DSPATR quando COLOR non è specificato esplicitamente

  - Supporto migliorato per l'override dei campi tramite la parola chiave REFFLD
  - Supporto migliorato per le specifiche di input per i record senza campi di input nel nuovo design
- PRTF
  - Il programma migliorato descriveva la trasformazione prtf con il file legacy .prtf vuoto
  - Impostazione del layout migliorata sulla generazione di modelli JASPER
  - Migliore gestione delle specifiche di AS400 output con modifica delle parole (modelli di formattazione)
  - Funzionalità di formattazione dei dati EditWord e gestione dei segni negativi implementate nelle specifiche di output RPGLE EditCode
  - È stato aggiunto il supporto per la parola chiave «DLTEDT»
  - È stato aggiunto il supporto per gli indicatori di output per i file PRTF
- Misc (Vari)
  - Supporto migliorato per la PFkey conversione per tenere conto della codifica durante la conversione delle costanti simboliche per le chiavi di aiuto CICS

AWS Transform for mainframe Transformation Engines & Runtime for GS21

Miglioramenti

- Lingue:
  - ADL: linguaggio di descrizione AIM, utilizzato per descrivere gli artefatti AIM (PEDs, schemi NDB (sub), schemi VSAM, procedure AIM, ecc.)
    - Trasformato in record di configurazione del database, script DDL, file JSON elaborati da Data Migrator, ecc...
  - ACSGEN: ACS Environment Generator, utilizzato per generare set di dati di sistema, tabelle di controllo, sorgenti ADL, punti di ingresso di programmi applicativi...
    - Trasformato in record di configurazione del database
  - PSAM: metodo di accesso al servizio di presentazione, utilizzato per descrivere le schermate online
    - Trasformato in artefatti angolari
  - JCLXSP: Job Control Language utilizzato sulla piattaforma GS21 XSP: sintassi speciale di JCL con funzionalità specifiche
    - Trasformato in script Groovy

  - GS21 COBOL: supporto per costrutti GS21 specifici (principalmente legati al supporto della lingua giapponese)
- Database:
  - NDB: database di rete
    - Trasformazione personalizzabile in un moderno database relazionale
    - Migrazione dei dati: elaborazione in formato GS21 TDUMP
    - Trasformazione dei collegamenti di rete in relazioni SQL moderne (utilizzando chiavi esterne e colonne aggiuntive per mantenere l' NEXT/PRIOR ordine derivante dal database precedente)
    - Generazione di codice di un moderno livello Java DAO supportato dal runtime appena introdotto GS21
- GS21 ecosistemi:
  - AIM: Advanced Information Manager. Il GS21 middleware principale gestisce schermi, stampanti, workstation, database e comunicazioni esterne tramite un approccio di messaggistica supportato da code.
  - ACSAPI: programma GS21 speciale che implementa il protocollo ACS che abilita scenari di visualizzazione (intercomunicazione tra programmi, interazioni, accesso VSAM, ecc...) PSAM/ printer (antenato di AIM).
  - OFM: Online Format Management, GS21 sottosistema utilizzato per la comunicazione di programmi tramite file di messaggi (antenato di ACS).
  - JXGIJSM, KQCAMS, JXKUNLOD, JXKRELOD KDJBR14, ecc...: varie utilità JCL GS21
  - Support per i comandi del sistema operativo
- Codifica specifica: JEF
  - Codifica a doppio byte (stessi principi IBM930 ma valori SOSI diversi (0x28, 0x29)).
  - Codifica JEF-PU personalizzata che preserva i caratteri per uso privato
  - Il carattere corrispondente viene utilizzato in per riprodurre il webapps/pdfs display con la massima fedeltà

Note di rilascio 4.10.0

Rilasciato il: 29 agosto 2025

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines introduce miglioramenti alle funzionalità della piattaforma, ottimizzazioni delle prestazioni e tecnologie modernizzate. Alcune funzionalità e modifiche chiave includono:

- Risoluzione migliorata della libreria AS/400

  È stato introdotto un sistema intelligente di gestione degli elenchi di librerie che offre il comportamento nativo degli elenchi di librerie AS/400 mantenendo prestazioni ottimali. Il sistema risolve automaticamente le librerie di oggetti in base all'elenco delle librerie del sistema (*LIBL) senza influire sulla velocità dell'applicazione.
- Modernizzazione del front-end

  Le applicazioni front-end sono state aggiornate alla versione 20 di Angular, che offre le funzionalità più recenti e i miglioramenti delle prestazioni all'interfaccia utente.

Questa versione di AWS Transform for mainframe Runtime è stata testata con il seguente stack:

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.17.1
	Npm 10.9.0
	Angolare 20
Livello di servizio	Spring Boot 3.4.6
	Spring Core 6.2.2
	Sessione primaverile 3.4.1
	Spring statemachine 4.0.0
Livello di persistenza	Motore PostgreSQL 15.10
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.40

AWS Transform per mainframe Runtime

ZoS

Miglioramenti

- COBOL
  - Supporto migliorato per la comunicazione dei parametri JCL-COBOL con una solida gestione a livello di byte per prevenire eccezioni ai limiti dell'array
  - Supporto migliorato per DFHEILBK e DFHEIBLC nelle istruzioni di copia
  - Supporto migliorato per la funzionalità dell'istruzione MOVE per gestire le conversioni di campi da non numerici a numerici con output previsto a valore zero
  - Supporto migliorato per l'elaborazione delle istruzioni EXAMINE con parametri di tipo intero
  - Aggiunto il supporto per l'utilizzo PROCEDURE-POINTER
  - Caratteri di escape XML migliorati
  - Aggiunto il supporto per l'istruzione FREE e l'istruzione Adapt ALLOCATE
- JCL - IDCAMS
  - È stato aggiunto il supporto per la parola chiave SKIP nell'istruzione REPRO
  - Supporto migliorato dell'istruzione DELETE per i file NONVSAM
- JCL - ICEGENER
  - Migliore gestione dei parametri DCB per dummy SYSUT1
  - Supporto migliorato per la gestione dei parametri Dynamic Control Block (DCB) quando SYSUT1 definiti come DUMMY, garantendo una corretta allocazione dei set di dati e prevenendo gli errori di elaborazione che potrebbero verificarsi durante le operazioni di copia con set di dati di input fittizi
- JCL - ORDINA
  - Aggiunto il supporto per il caso di file vuoti concatenati
  - Migliore confronto dei record aggiornando il calcolo dei byte per i campi ZD (Zone Decimal)
  - Supporto migliorato per JOINKEY - FILL e REFORMAT FIELD
  - Elaborazione dei file concatenati migliorata per mantenere correttamente la coerenza delle dimensioni dei record tra i set di dati di input
  - Grammatica di ordinamento migliorata e gestione del runtime corrispondente per X'.. ' (lettere esadecimali), DATE =(...) e parole chiave TIME utilizzate nelle clausole HEADER e TRAILER di OUTFIL.

- JCL - ICETOOL
  - Aggiunto il supporto per alcune opzioni dell'operatore COUNT: EMPTY, NOTETEMPTY, HIGHER (x), LOWER (y), EQUAL (v), NOTEQUAL (w)
- JCL - DSNUTILB
  - Supporto migliorato per l'inserimento di colonne DATE nulle
  - È stato aggiunto il supporto per i tipi di dati VARCHAR, DECIMAL e INTEGER quando si utilizzano i delimitatori
  - Supporto migliorato per l'inserimento di valori predefiniti quando le colonne non sono specificate nelle schede di controllo
  - Supporto migliorato per l'elaborazione di record a lunghezza variabile escludendo i campi RDW (Record Descriptor Word) dal contenuto del record
  - Supporto migliorato per la gestione dei valori nulli nelle colonne del database durante le operazioni LOAD
  - Supporto migliorato per il caricamento di tipi di dati decimali per garantire conversioni numeriche precise
  - Supporto migliorato per il parametro WORKDDN se utilizzato in combinazione con le specifiche FORMAT DELIMITED e POSITION (*)
- JCL - Varie
  - Aggiunto il supporto per SystemOut le definizioni con sintassi basata su virgole come SYSOUT =(,)
  - Supporto migliorato per le istruzioni DD contenenti il carattere speciale #
  - Supporto migliorato per la gestione di più condizioni tra parentesi chiuse
  - Aggiunto il supporto per la sintassi DSN racchiusa tra virgolette
  - È stato aggiunto il supporto per la specifica della sintassi COPYDDN (nome)
  - Supporto migliorato per le istruzioni DD duplicate mantenendo un comportamento simile a z/OS di mantenimento della prima assegnazione
  - Supporto migliorato per la gestione dei set di dati SYSOUT con i parametri SPIN e FREE, che consente il buffering e la stampa corretti del contenuto SYSOUT DD end-of-job
  - Segnalazione degli errori migliorata per i lavori nell'API di esecuzione Groovy
  - Attivata l'allocazione dinamica di file temporanei tramite una proprietà YML
- Blusam

- Supporto migliorato per le operazioni CLEAR migliorando i processi di pulizia dei metadati per garantire la completa rimozione di tutti i riferimenti ai dati precedentemente archiviati
- Prestazioni migliorate delle operazioni WRITE con la modalità write-behind abilitata
- Supporto migliorato delle operazioni sui set di dati ottimizzando i processi di verifica dell'esistenza delle tabelle, eliminando le operazioni di ricerca ridondanti e migliorando le prestazioni complessive grazie alla riduzione delle query sul database
- Prestazioni migliorate aggiungendo una nuova modalità aperta (modalità solo blocco a livello di set di dati) che imposta la modalità di inserimento in blocco su true. Questa configurazione è utile negli scenari di ripristino di set di dati personalizzati in cui i blocchi a livello di set di dati sono sufficienti.
- Configurazione estesa per <noloc>Blusam</noloc> KSDS di grandi dimensioni, che consente l'uso del file di configurazione yml ds per impostare le proprietà
- Aggiunto il supporto per la grafica alfanumerica nella generazione di maschere
- Segnalazione degli errori migliorata quando l'indice alternativo non è presente nei metadati del set di dati
- È stata abilitata l'esecuzione asincrona delle operazioni di scrittura in blocco di Redis per migliorare le prestazioni. La funzionalità è configurabile tramite una proprietà YML.

- SQL
  - È stato aggiunto il supporto per l'istruzione CONNECT
  - Supporto SQL migliorato per i nomi delle variabili e le interrogazioni UPDATE delle espressioni degli operatori che coinvolgono cursori
  - Migliore sostituzione delle stringhe nell'istruzione SQL Prepare
- Sicurezza
  - Maggiore sicurezza delle API: implementata l'autenticazione obbligatoria e l'autorizzazione basata sui ruoli per tutti gli endpoint REST delle applicazioni gapwalk quando la configurazione di sicurezza è abilitata.
  - Autorizzazione migliorata dei token JWT implementando una scope-to-role mappatura personalizzata, che consente l'accesso adeguato agli endpoint protetti quando si utilizzano token bearer emessi da Cognito*.
- Misc (Vari)
  - Integrazione Redis migliorata con registrazione, gestione degli errori e diagnostica complete, inclusi oltre 50 nuovi controlli contestuali e 137 messaggi di errore perfezionati.
  - Migliore inizializzazione del job alla prima transazione in esecuzione

AS400

Nuove funzionalità

- È stato aggiunto il supporto per la funzionalità di elenco delle librerie per automatizzare la risoluzione delle librerie in base alle librerie disponibili e all'elenco delle librerie
- CL
  - È stato aggiunto il supporto per il comando QDBRTVFD che recupera la descrizione di un file di database
  - È stato aggiunto il supporto per il comando QCMDCHK per il controllo e l'analisi.

Miglioramenti

- RPG
  - Supporto migliorato del funzionamento Z-ADD per gestire il caso di aggiunta di 2 array di tipo Packed
  - Supporto migliorato per l'inizializzazione costante con funzioni integrate
  - Supporto migliorato di EDTWRD con una stringa vuota
  - Supporto migliorato dell'istruzione MOVE per gestire il caso di valori negativi assegnati a campi compressi senza parti intere
  - Supporto migliorato della parola chiave *LOVAL per i tipi Character e Zoned
- COBOL
  - Supporto migliorato dello stato del file COBOL per gestire i casi in cui il campo non è un intervallo elementare
  - Migliore impostazione del codice di stato del file al momento del recupero in modalità Random
  - Codifica delle stringhe migliorata per la funzione COBOL DISPLAY-OF
- CL
  - Aggiunto il supporto per il comando QCMDCHK
  - Supporto migliorato per il comando CPYF per gestire il formato multi-record LFs
  - Supporto migliorato per il comando RMVM per eliminare un membro quando altri oggetti dipendono da esso
  - Supporto migliorato dei parametri per gestire gli spazi finali nel comando RMVM
  - Comando QCMDEXC migliorato con una migliore gestione degli argomenti, incluso il supporto per i parametri posizionali e il corretto recupero dell'ordine degli argomenti

  - Comando OVRPRTF migliorato implementando il valore predefinito (*STD) per il parametro FORMTYPE non specificato
  - Supporto migliorato per RTVJOBA per la restituzione del numero di lavoro con 6 caratteri
  - Supporto migliorato per il comando OPNQRYF e il parametro KEYFLD
  - Supporto migliorato per i caratteri Unicode per il comando OPNQRYF e l'escape tra virgolette singole
  - Migliore gestione del nome del file di output per il comando CPYF
  - Comando RTVMBRD migliorato con funzionalità avanzate di recupero delle visualizzazioni, incluso il supporto per le operazioni senza condizioni where e una corretta gestione dell'intestazione per il conteggio delle righe
  - Supporto migliorato per il comando DSPJOB per gestire le chiamate utilizzando i parametri PATH e JOB
  - Supporto migliorato per il comando RTVOBJD per gestire i file nella directory
  - Supporto migliorato del comando ADDPFM per gestire i parametri contenenti spazi finali
  - Migliore sincronizzazione dei thread in CHGDTAARA
- Accesso al database
  - Funzionalità avanzate di sovrascrittura dei file per supportare i file JOIN
  - Regole di precedenza avanzate degli AND/OR operatori durante l'elaborazione di più condizioni Select/Omit
- Schermo
  - Posizione del cursore migliorata quando due schermate condividono il nome di un campo
  - Supporto migliorato del campo RRN per il file Workstation
  - È stata gestita la schermata dei messaggi aggiuntivi da Modal windows
  - Visualizzazione migliorata della riga del messaggio come matrice in formato DS4
  - È stata migliorata la visualizzazione dei messaggi di errore e la schermata di informazioni aggiuntive sui messaggi AS400 nelle applicazioni
  - Supporto migliorato del carattere @ nel front-end modernizzato
  - Migliore gestione delle sessioni nel front-end per consentire l'esecuzione di più sessioni sullo stesso browser con più schede
- Stampante
  - Migliore gestione dell'overflow per i file di stampante con più intestazioni di record.
- Misc (Vari)

- Aggiunto il supporto Redis per gli spazi utente
- È stato aggiunto il supporto per l'API QUSLFLD che genera un elenco di campi all'interno di un nome di formato di record di file specificato utilizzando lo spazio utente.

AWS Transform per motori di trasformazione mainframe

ZoS

Miglioramenti

- COBOL
  - Supporto migliorato per CLOB-LOCATOR/DBBLOB-LOCATOR/BLOB-LOCATOR
  - Supporto migliorato per l'istruzione IF con condizioni binarie abbreviate
  - Supporto migliorato per la DFHEIBLK/EIBLK gestione con e senza istruzioni di copia
  - Funzionalità raffinata della macchina a stati con gestione avanzata dei commenti e logica EXIT ottimizzata per chiamate a più sezioni.
  - Supporto migliorato per l'istruzione EXIT SECTION
  - Supporto migliorato della parola REPLACING ALL nell'istruzione INITIALIZE
- SQL
  - Supporto migliorato per la dichiarazione di alias SQL per la query di aggiornamento
  - Supporto migliorato per la creazione di partizioni PostgreSQL ignorando le parole chiave ENDING e INCLUSIVE
- Informazioni
  - Segnalazione degli errori di trasformazione migliorata

AS400

Miglioramenti

- DSPF
  - Calcolo della lunghezza del campo migliorato per corrispondere con precisione al formato dei valori, evitando problemi di visualizzazione
- PRTF
  - Migliore generazione del numero di pagina nei report composti da più pagine

    - Visualizzazione costante migliorata nel report
- RPG
    - Aggiunto il supporto per *ALL quando utilizzato con Z-ADD su un array
    - Supporto migliorato della costante figurativa *ALL con tipi esadecimali
    - È stato aggiunto il supporto per *HIVAL quando specificato come parametro per l'opzione INZ
    - Ha gestito l'extender H per arrotondare il risultato dell'operazione DIV
    - Supporto migliorato per la funzione LOOKUP per gestire gli indicatori alti, bassi e uguali in base al tipo di array di input passato (ASCENDENTE O DISCENDENTE)
    - Supporto migliorato per la funzione ADD per eseguire l'aggiunta per elemento quando i campi factor1, factor2 e result sono tutti array
    - Supporto migliorato per la funzione EDTWRD nella dichiarazione dei campi
    - Supporto migliorato del funzionamento CLEAR con il file INPUT
    - Migliore inizializzazione degli array definiti come campi autonomi
    - Supporto migliorato dell'opzione*NEXT nella parola chiave OVERLAY
    - Supporto migliorato per la parola chiave *ZEROS quando viene utilizzata nell'operazione MOVEA
- Misc (Vari)
    - Supporto migliorato per numeri di grandi dimensioni utilizzando BigDecimal

Note di rilascio 4.9.0

Data di rilascio: 17 luglio 2025

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines introduce aggiornamenti chiave alle dipendenze principali:

- AS400: Abbiamo introdotto un'implementazione alternativa basata su JDBC oltre al supporto JPA esistente nel nostro framework DAO. Gli utenti possono ora passare tra le implementazioni JPA e JDBC tramite la configurazione YML. I benchmark iniziali mostrano che il nuovo supporto JDBC offre significativi miglioramenti delle prestazioni, riducendo i tempi di esecuzione di circa il 50% rispetto all'implementazione JPA.
- zOS: introdotti gli ambienti a e LUW. DB2 z/OS to DB2 LUW migration solution addressing the subtle but critical syntax differences between DB2 z/OS DB2 Sebbene le sintassi dei database appaiano simili, i relativi requisiti specifici richiedono una gestione automatizzata delle

trasformazioni per la gestione degli schemi, gli spazi delle tabelle e i tipi di dati. Ciò garantisce il successo della migrazione dalle piattaforme mainframe a quelle distribuite, mantenendo al contempo l'integrità operativa.

Abbiamo testato questa versione del AWS Transform for mainframe Runtime con lo stack seguente.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.11.0
	Npm 10.9.0
	Angolare 19.1.3
Livello di servizio	Spring Boot 3.4.6
	Spring Core 6.2.2
	Sessione primaverile 3.4.1
	Spring statemachine 4.0.0
Livello di persistenza	Motore PostgreSQL 15.10
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.40

Versione di runtime 4.9.0

ZoS

Miglioramenti

- COBOL
  - Aggiunto il supporto per la funzione DATE-TO-YYYYMMDD
  - Supporto migliorato per l'istruzione UNSTRING per migliorare il calcolo TALLYING

  - Aggiunto il supporto per il valore DFHRESP (NORMAL) sulla variabile 88 Level
  - Operazioni di associazione migliorate che coinvolgono COMMAREA, in particolare sulle chiamate di programma annidate
  - Aggiunto il supporto per i registri speciali LINE-COUNTER e PAGE-COUNTER
  - Supporto migliorato per l'istruzione WRITE... BEFORE per il file Record Sequential in base ai caratteri di controllo automatico per i canali (da C01 a C12)
  - Supporto migliorato per le istruzioni che coinvolgono il file READ/WRITE variable-length-record
  - Supporto migliorato per l'istruzione CALL quando si passa un parametro per riferimento con una differenza di dimensione al sottoprogramma
  - Supporto SORT KEY migliorato quando la struttura ha 88 campi di livello
  - Supporto migliorato per il valore intero con segno nell'istruzione DISPLAY
  - Miglioramento delle prestazioni delle operazioni di concatenazione di stringhe grazie alla riduzione del sovraccarico di calcolo
  - Supporto migliorato per il calcolo MOD e REM per gestire valori negativi

- PL/I
  - Supporto migliorato della funzione SUBSTR con operazioni sui bit

- JCL - IDCAMS
  - Supporto migliorato per i file GDG per l'ordine di copia
  - È stato aggiunto il supporto per sovrascrivere qualsiasi specifica a livello di cluster quando KEYS è definito nel componente DATA
  - È stato aggiunto il supporto per l'istruzione ALTER NEWNAME
  - È stato aggiunto il supporto per la parola chiave opzionale GDG per le istruzioni DELETE
  - Gestione migliorata del codice delle condizioni assicurando che le sostituzioni MAXCC persistano tramite i comandi di controllo (SET, IF-THEN-ELSE) mentre vengono ripristinate in modo appropriato prima dei comandi funzionali

- JCL - ORDINA
  - Supporto migliorato per la gestione dei campi ZD con una combinazione di dati compressi e numerici

  - Supporto migliorato per il confronto decimale suddiviso in zone con Character Field nell'istruzione INCLUDE
  - Supporto migliorato per INREC FINDREP per la gestione delle parentesi nel parametro OUT
  - Supporto migliorato per l'istruzione OMIT per le condizioni ALL e NONE
  - Supporto migliorato per l'istruzione END
  - Supporto migliorato per l'ordine di precedenza degli operatori logici per la frase COND
  - Supporto migliorato per UNPAIRED in JOINKEYS

- JCL - DSNUTILB
  - È stato aggiunto il supporto per il tipo di dati DOUBLE PRECISION
  - Supporto migliorato per LOAD DATA con parola chiave COLDEL e usi di delimitatori speciali regex
  - Supporto migliorato dell'istruzione TEMPLATE for LOAD

- JCL - Varie
  - Supporto migliorato per troncare il record di Fixed Block Output basato su LRECL
  - Eliminazione dei file migliorata per prevenire problemi in ambienti ad alta disponibilità
  - È stato aggiunto il supporto per l'utilità di sistema IKJEFT1 B
  - Supporto migliorato per mantenere le dimensioni dei record in un JOB in base ai passaggi eseguiti in precedenza
  - Supporto migliorato per PARM e PARM incentrato su STEP in JCL e PROC
  - Supporto migliorato per la conformità di DCB=*.DD al Backward Reference utilizzato per fare riferimento alle informazioni da un DD definito in precedenza
  - Supporto migliorato per rendere disponibili le variabili della procedura di elaborazione in un contesto multithreading

- <noloc>Blusam</noloc>
  - È stata migliorata la lettura di <noloc>Blusam</noloc> Large KSDS durante la navigazione verso i record successivi/precedenti ai limiti della pagina o dell'indice
  - <noloc>Sono stati migliorati la cancellazione e il caricamento di file Blusam di grandi dimensioni</noloc>

  - Supporto migliorato per la creazione di tabelle di indici per <noloc>Blusam</noloc> Large KSDS quando manca un set di dati opzionale
  - Implementazione Write-Behind di <noloc>Blusam</noloc> migliorata per risolvere i problemi relativi alle operazioni di eliminazione e imporre l'accodamento degli aggiornamenti in batch
  - Prestazioni migliorate di <noloc>Blusam</noloc> Large data sets (aggiunta ed eliminazione) delle operazioni di registrazione (aggiunta ed eliminazione) utilizzando un meccanismo di memorizzazione nella cache degli indici locali che riduce i ritardi dovuti a interazioni multiple con la cache remota

- SQL
  - Supporto migliorato per le dichiarazioni con HAVING e senza clausola GROUP BY
  - Meccanismo di memorizzazione nella cache dei metadati degli schemi di database migliorato

- Integrazione della coda dei messaggi
  - Supporto migliorato per il funzionamento MQ CLOSE quando il lavoro viene eseguito con un approccio multithread
  - Prestazioni migliorate grazie all'implementazione del JMSTemplate riutilizzo per MQ Operations

- MFS
  - Supporto avanzato per il parametro DSCA nell'istruzione MFS DEV

- Schermo
  - Meccanica di immissione dei dati migliorata per impedire l'inserimento di caratteri a byte singolo in campi a doppio byte, specialmente quando si utilizza una tastiera virtuale, ad esempio il sistema di composizione del testo.

- Misc (Vari)
  - Sono state aggiunte altre opzioni di configurazione per il pool di connessioni Redis, ad esempio,, e `connectionPoolSize` `connectionMinimumIdleSize` `idleConnectionTimeout` `connectTimeout`
  - Endpoint /triggerscript/ {scriptID: .+} migliorato per supportare le richieste POST con payload JSON, che consente il passaggio di parametri complessi in aggiunta ai parametri di query esistenti

AS400

Miglioramenti

- RPG
  - È stato migliorato il supporto per gli `L0` indicatori nei calcoli totali
  - Migliorato il supporto dell'opcode TIME tramite l'inizializzazione del lavoro
  - È stato migliorato DaoCycleManager il modo di leggere i file primari e secondari con un numero diverso di campi di confronto
  - Formattazione numerica %EDITCODE 'J' migliorata
  - È stato migliorato il supporto del raggruppamento del codice di modifica per il codice di modifica 4

- COBOL
  - Supporto migliorato di REWRITE in modalità dinamica quando la clausola WITH DUPLICATE è specificata nel file RECORD KEY
  - Abilita l'aggiornamento dei campi chiave in modalità dinamica con la parola chiave DUPLICATES
  - Migliore gestione delle risorse nell'operazione SORT

- CL
  - Supporto migliorato per `CHKOBJ` il comando per gestire gli spazi finali nei nomi degli oggetti
  - Supporto migliorato per `CHKOBJ` il comando per verificare l'esistenza di una vista con una libreria
  - È stata migliorata l'analisi degli elenchi di parametri per il comando `SBMJOB`
  - Supporto migliorato per `CLRSVF` il comando e `MSGMON` per generare codice `CPF9812` quando il file non è esistente
  - Supporto migliorato per `OVRDBF` il comando di gestione `*END` del valore per il parametro `POSITION`
  - Supporto migliorato per l'`*REPLACE`accesso `CPYF` quando il file sorgente è vuoto
  - Supporto migliorato per il `CHGVAR` comando di gestione dei parametri senza la parola chiave opzionale VAR

- Accesso al database
  - Supporto migliorato dell'operazione DELETE per lavori simultanei

  - Supporto migliorato dell'operazione DELETE come prima operazione su un file.
  - Migliore integrità dei dati durante la lettura da file join
  - Migliore gestione dei caratteri di controllo READ/WRITE nelle operazioni quando il flag INSERTNULL è disattivato.
  - Aggiunte le colonne tecniche SRCFILE e MEMBER nelle viste SQL per fornire meccanismi per gestire le librerie
  - Miglioramento del calcolo del timeout residuo nei blocchi degli oggetti
  - Supporto migliorato per l'override dei file sui file JOIN
  - Aggiunto il supporto per Relative Key on Transaction File

- Schermo
  - Migliorata la gestione di più messaggi di errore per lo stesso campo
  - È stata gestita la visualizzazione dei messaggi STATUS inviati alla coda di messaggi esterna del processo
  - Aggiunto il servizio Message Line per controllare dinamicamente la visualizzazione dei messaggi di errore
  - Migliorata la risoluzione del messaggio per la schermata aggiuntiva

- Misc (Vari)
  - È stato migliorato il meccanismo di tracciamento della sessione per restituire il timestamp dell'ultimo aggiornamento e il timestamp di creazione formattati correttamente
  - Migliore coerenza in tutte le raccolte di record quando i record sono nel database updating/deleting
  - È stata migliorata l'inizializzazione degli switch as400 all'avvio del job introducendo una nuova proprietà YML `job.default.encoding` che specifica la codifica predefinita utilizzata per inizializzare la memorizzazione a livello di processo con default = `CP1047`
  - Grammatica SQL migliorata per accettare la sequenza di escape tra virgolette
  - Supporto migliorato per i tipi di chiavi compressi per OVRDBF POSITION.

Motore di Transform for mainframe trasformazione AWS 4.9.0

ZoS

Nuove funzionalità

- SQL
  - Aggiunto il supporto per DDL e la trasformazione delle query in linea da zOS a LUW DB2 DB2

Miglioramenti

- COBOL
  - Supporto migliorato: campi di livello 88 con campo principale di livello 01
  - Supporto migliorato per i nomi speciali `CTL` e l'`WRITE .... AFTER CTL`istruzione
  - È stato migliorato il passaggio da un paragrafo all'altro durante `PERFORM SECTION`
  - È stato aggiunto il supporto per l'`COPY`istruzione con percorso relativo
  - È stato aggiunto il supporto per `BODY` un nome di campo COBOL valido
  - Supporto migliorato per la dichiarazione `EVALUATE ... WHEN`
  - Supporto migliorato per la `RECORDING MODE` clausola nel file `SD`
  - È stato aggiunto il supporto per le costanti figurative nella dichiarazione `WRITE`
  - È stato aggiunto il supporto per S01 TO S05 e C01 TO C12 come nomi di campo validi
  - È stato aggiunto il supporto `PRINTER` come file di destinazione per la clausola `SELECT ASSIGN`
  - È stato aggiunto il supporto per l'istruzione REPLACE nella clausola `FD`
  - È stato aggiunto il supporto per la riga di definizione dei dati `COPY` nella clausola `FD`
  - È stato aggiunto il supporto per l'`COPY`istruzione con percorso relativo
  - Supporto migliorato per l'`STRING`istruzione per supportare l'elemento dell'array come delimitatore, ad esempio. `STRING ABCDEFG DELIMITED BY IDX-1 (SUB) INTO IDX-2 POINTER ID3`
  - Supporto aggiunto per il `DFHVALUE` valore su una variabile di 88 livelli per una struttura di dati
  - Supporto migliorato per l'`FUNCTION LENGTH`istruzione quando l'argomento è un'espressione SUBSTRING
  - È stato aggiunto il supporto per la modalità LOCK Close
  - Aggiunto il supporto per OF/IN nell'istruzione START

- Supporto migliorato per la clausola JUSTICIED quando la parola chiave RIGHT non è specificata
- Supporto migliorato per la denominazione FD per evitare collisioni per parole chiave riservate
- Supporto migliorato per i nomi speciali con clausola condizionale
- Aggiunto il supporto per DEBUGGING MODE nella configurazione SOURCE-COMPUTER
- Supporto migliorato per i paragrafi opzionali IDENTIFICATION DIVISION e DATA WRITTEN
- Stati di uscita sicuri migliorati nelle sezioni dichiarative
- È stato aggiunto il supporto per le istruzioni chiave non valide per i file sequenziali
- Supporto migliorato per i nomi duplicati dalle sezioni WORKING STORAGE e LINKAGE
- Supporto migliorato dell'istruzione SET ADDRESS OF DFHCOMMAREA
- Supporto migliorato per la chiamata dichiarativa COBOL

- PSB
  - Supporto migliorato per la `SENSEG` dichiarazione where `PROCOPT` is before `PARENT`

- JCL
  - IDCAMS - Aggiunto il supporto per le abbreviazioni REPL, IMBD e WCK rispettivamente per REPLICATE, IMBED e WRITECHECK

- SQL
  - Supporto migliorato per l'espressione di tabelle annidate

AS400

Miglioramenti

- CL
  - Supporto migliorato dei tipi di campo durante CL-to-groovy la generazione
- DDS
  - La parola chiave DPSF RTNCSRLOC supporta più parametri di separazione degli spazi
  - Determinazione del nome del formato di record migliorata per gli indici in SQL-DDS.
  - Supporto migliorato per la sequenza di escape delle citazioni all'interno di SQL-DDS. `' '`
- RPG

- RPG
  - Supporto migliorato per le `LEAVE` istruzioni all'interno di un ciclo riscritto per supportare GOTO
  - È stato aggiunto il supporto per la funzione `%TIMESTAMP` integrata utilizzata in una chiamata di procedura
  - Supporto migliorato per la parola chiave F-Spec contenente lettere maiuscole e minuscole
  - Supporto migliorato per il codice controllato dagli indicatori del livello di controllo
  - Supporto migliorato per gli indicatori di arresto
  - Supporto TAG migliorato con Control Level
  - Supporto migliorato per le `CLEAR` dichiarazioni mediante la gestione dei dati ricorrenti nei layout di record esterni
  - Supporto migliorato degli array di indicatori
  - Supporto migliorato dell'array CTDATA
  - Supporto migliorato per l'assegnazione di array fissi elementari
  - Supporto migliorato per espressioni di funzioni annidate complesse nei calcoli RPG
  - read/write Generazione avanzata di aree dati locali con UDS
- SQL
  - Supporto avanzato per la sintassi SQL con parentesi nella clausola FROM.

Note di rilascio 4.8.0

Data di rilascio: 23 aprile 2025

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines introduce aggiornamenti chiave per migliorare le funzionalità e le prestazioni di trasformazione del database:

- Trasformazione del database IBM IMS: è stato aggiunto il supporto per la trasformazione dei database IBM IMS nella nostra soluzione personalizzata JHDB. AWS Transform for mainframe
- Funzionalità di cache in memoria: è stata aggiunta la funzionalità di cache in memoria che consente agli utenti di memorizzare nella cache i dati di sola lettura all'interno della memoria, migliorando le prestazioni per i programmi ad uso intensivo di dati.

Abbiamo testato questa versione di AWS Transform for mainframe Runtime con lo stack seguente.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.11.0
	Npm 10.9.0
	Angolare 19.1.3
Livello di servizio	Spring Boot 3.4.2
	Spring Core 6.2.2
	Sessione primaverile 3.4.1
	Spring statemachine 4.0.0
Livello di persistenza	Motore PostgreSQL 15.10
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

Versione di runtime 4.8.0

ZoS

Nuove funzionalità

- È stato introdotto il supporto per la definizione di intestazioni personalizzate per transazioni protette tramite la proprietà YML. `gapwalk-application.security.customAllowedHeaders` Questa proprietà è correlata alla `gapwalk-application.identity` proprietà con valore. `oauth`
- <noloc>È stata aggiunta una funzionalità per consentire ai clienti di ricostruire i metadati <noloc>Blusam</noloc> in base ai record effettivi nella tabella del database Blusam</noloc>
- Gestione dei segreti AWS Connessioni HDB protette tramite integrazione

Miglioramenti

- COBOL
  - Supporto migliorato per la condizione AT END sull'istruzione del file READ
  - Supporto migliorato per SPOSTARE i letterali numerici firmati in campi alfanumerici senza spostare il segno
  - Supporto migliorato per l'istruzione UNSTRING per gestire le conversioni di campi da non numerici a numerici e mantenere valori bassi quando il delimitatore non viene trovato e nessun valore viene spostato in altre variabili
  - È stata migliorata l'operazione di associazione degli indirizzi per le entità di collegamento dinamico
  - Operazioni di concatenazione di stringhe migliorate grazie all'inclusione del supporto per shift bytes (SOSI) in/out
  - Supporto migliorato per l'istruzione COBOL ACCEPT con sorgente DAY-OF-WEEK
  - È stato aggiunto il supporto per SYSIN e SYSOUT come valori SPECIAL NAMES
  - Supporto migliorato per la concatenazione di STRING per gestire il carattere del form feed
- JCL - IDCAMS
  - Aggiunto il supporto per l'eliminazione di generazioni gdg utilizzando un nome base senza il parametro GDG
  - Supporto migliorato per la gestione del parametro VOLUME come variante del parametro VOLUMES standard per l'istruzione DEFINE CLUSTER
- JCL - ORDINA
  - Aggiunto il supporto per SYMNAMES con il comando JOINKEYS
- JCL - IEBGENER
  - Supporto migliorato per IEBGENER per la gestione di contenuti SYSIN non validi introducendo controlli di convalida e fallback automatico su ICEGENER, garantendo un funzionamento continuo anche con istruzioni di controllo errate
- JCL - INFUTILB
  - Migliore gestione delle clausole unload INTO e FROM
- JCL - DSNUTILB
  - Aggiunto il supporto per il tipo di dati DOUBLE PRECISION
- JCL - Varie
  - È stata migliorata la sovrascrittura delle configurazioni dei file utilizzando i nomi DD nella procedura secondaria del job

  - DSNTIAUL/INFUTILB - Personalizzazione migliorata dei byte di riempimento SYSREC per l'utilità unload
  - Aggiunto il supporto per l'opzione OUTDD per il reindirizzamento dell'istruzione DISPLAY
  - Supporto migliorato per la scrittura di file sequenziali a blocchi fissi. Quando il LRECL nel programma COBOL è diverso dal JCL LRECL, il JCL LRECL ha la precedenza.
  - È stato aggiunto il supporto per gli attributi DCB del catalogo per il file SORTOUT, ma solo per i set di dati permanenti
  - Supporto migliorato per la trasformazione condizionale dei passaggi JCL
  - Supporto migliorato per ACCEPT con SYSIN e linee che superano la dimensione desiderata con spazi finali
  - Supporto migliorato per le istruzioni SET per gestire comandi che non includono il nome della variabile, il comando definito all'interno include un file membro che dovrebbe essere accessibile all'interno di JCL e il comando contiene commenti in linea
  - È stato aggiunto il supporto per conservare i parametri 'ScriptContext' a livello di lavoro nei metadati del checkpoint JCL e il contesto del lavoro da utilizzare al riavvio
- <noloc>Blusam</noloc>
  - Inserimento dei record migliorato su grandi set di dati indicizzati <noloc>Blusam</noloc> durante la scrittura dopo il caricamento di un set di dati vuoto
  - Prestazioni migliorate per set di dati di grandi dimensioni abilitando un meccanismo di riscaldamento e introducendo una finestra di prefetch opzionale per i record, nonché l'archiviazione locale per indici e pagine
  - Supporto migliorato per Export Data Set per gestire un set di dati più grande con AWS Key Management Service
- SQL
  - Supporto migliorato per il tipo di dati JDBC BLOB
- Integrazione della coda dei messaggi
  - Aggiunto il supporto per l'identificatore utente sul comando MQ PUT quando XA è attivato
  - Aggiunto il supporto per connessioni MQ simultanee quando XA è attivato
- IMS
  - È stata aggiunta una proprietà YML `jhdb.keepParent` per controllare se le chiamate IMS ISRT popolano i campi _parent e _logicalparent nel database quando sono impostati su true
  - Maggiore robustezza rimuovendo la coda dei messaggi per il codice di transazione non valido

AS400

Nuove funzionalità

- È stata aggiunta la funzionalità di cache in memoria che consente agli utenti di memorizzare nella cache i dati di sola lettura all'interno della memoria, migliorando le prestazioni per i programmi a uso intensivo di dati.

Miglioramenti

- CL
  - Supporto migliorato del parametro QRYSLT nel comando OPNQRYF per gestire la parola chiave RANGE, analizzare il valore quando le colonne di stringhe hanno valori puramente numerici e gestire valori di stringa vuoti
  - Aggiunto il supporto per il comando FTP SENDPASV
  - Supporto migliorato per la formazione di query nel comando DSPDBR eseguito utilizzando Qcmdexc
  - È stato migliorato il supporto dell'eccezione MONMSG per propagarla correttamente nello stack di esecuzione
  - È stata migliorata la copia dei record tra tabelle con conteggi di colonne e tipi di dati diversi su CPYF
  - Migliore priorità di monitoraggio dei messaggi: i messaggi a livello di comando ora hanno correttamente la precedenza su quelli a livello di procedura
  - Supporto migliorato per il comando CPYF per considerare la libreria definita da IVLIB BLU4
  - Supporto migliorato del comando MonMsgs in CHKOBJ per gestire il caso in cui il file non esiste nella libreria
- Schermo
  - Migliore gestione dei BigDecimal valori con più di 15 cifre costruendo il valore manualmente e quindi trasformandolo in un numero da visualizzare anziché arrotondare il numero
- Misc (Vari)
  - Supporto migliorato per UserSpace Replace as NO quando UserSpace già esiste
  - Funzionalità BLU4 IVLIB migliorata per supportare sia il recupero che la scrittura di record in librerie sostituite

- Supporto migliorato per i campi di matrice multidimensionale utilizzando le operazioni DIM e OCCURS
- Supporto migliorato per valori bassi passati come valori di campo chiave nel funzionamento DAO
- È stato rimosso il conflitto nell'istruzione MONMSG per i messaggi che possono essere gestiti da un monitor locale e globale
- Impostazione migliorata FileStatus su Duplicate Record Insert
- Sono stati ridotti i conflitti di blocco dello spazio utente e il rischio di errori di creazione utilizzando innanzitutto il registro per verificare l'esistenza dello spazio utente
- Supporto migliorato per readPriorEqual () saltare i record dopo delete ()
- Meccanismo di blocco dei record migliorato sotto il controllo degli impegni
- Migliore gestione dell'istruzione RPG RETURN nelle subroutine
- È stato migliorato il supporto delle informazioni utente in SharedContext

Motore di Transform for mainframe trasformazione AWS 4.8.0

ZoS

Miglioramenti

- COBOL
  - Generazione della modalità stampante migliorata per più file.
  - Supporto migliorato per SPECIAL-NAMES per gestire punti finali e punti multipli opzionali.
  - Supporto migliorato di COPY con il percorso relativo alle istruzioni /REPLACE OFF/COPY-REPLACING.
  - È stato aggiunto il supporto per COL come nome di campo valido.
  - Analisi migliorata per gestire gli spazi dopo SPECIAL-NAMES, X-COMPUTER e DECLARATIVES.
  - È stato aggiunto il supporto per i nomi di paragrafo da S01 a S05 e da C01 a C12 come nomi di paragrafo validi.
  - È stato aggiunto il supporto per la funzione LENG come alias della funzione LENGTH.
  - Analisi migliorata dei Copybook quando il valore di configurazione cobolMaxCode Width è diverso dal relativo inclusore COBOL
  - Supporto migliorato per l'analisi di espressioni con virgole e operazioni, ad esempio MIN (A* B, (3 + 1)/2, 3 + 4).

  - È stato aggiunto il supporto per IS INITIAL PROGRAM nella clausola PROGRAM ID.
  - È stato aggiunto il supporto per la condizione di livello 88 nell'istruzione SEARCH ALL.
  - È stato aggiunto il supporto per REPORT SECTION.
- SQL
  - È stato aggiunto il supporto per la trasformazione DDL dal database IBM IMS.

AS400

Miglioramenti

- CL
  - Supporto migliorato di CL impostando la trasformazione in JAVA come supporto predefinito
  - È stata migliorata la trasformazione in script groovy aggiungendo l'uso di un metamodello
  - Aggiunte le parole chiave CONTINUE e FOR all'elenco delle parole chiave riservate
- DDS
  - Supporto migliorato per i campi DSPF definiti rispetto a quello precedente
  - Supporto migliorato per il campo di caratteri DDS con la parola chiave VARLEN
  - Miglioramento transformation/generation dell'entità LF DAOs generando l'impostazione esplicita dei campi rinominati nei metodi di conversione e aggiornamento Vo. JPAFrom
- RPG
  - Supporto migliorato del parametro della parola chiave PREFIX per gestire i casi in cui i valori non hanno virgolette
  - Migliore gestione delle chiamate ai metodi annidati negli indici degli array
- COBOL400
  - Migliorata la generazione di entità record importate da un DSPF con la direttiva COPY DDS-ALL-FORMATS
- SQL
  - È stata migliorata la trasformazione delle clausole SQL WHERE con condizioni OR non racchiuse.

Note di rilascio 4.7.0

Data di rilascio: 12 marzo 2025

Questa versione di AWS Transform for mainframe Runtime e Transformation Engines introduce aggiornamenti chiave alle dipendenze principali: AWS Transform for mainframe Runtime è stato aggiornato per utilizzare l'ultima versione Spring Boot 3.4.2 e le applicazioni frontend modernizzate ora utilizzano l'ultima versione 19 di Angular.

Abbiamo testato questa versione di Transform for mainframe Runtime con il AWS seguente stack. Potrebbero essere compatibili anche altre versioni.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.11.0
	Npm 10.9.0
	Angolare 19.1.3
Livello di servizio	Spring Boot 3.4.2
	Spring Core 6.2.2
	Sessione primaverile 3.4.1
Livello di persistenza	Spring State Machine 4.0.0
	Motore PostgreSQL 14
Oracle 21c	
Server applicazioni	Apache Tomcat 10.1.17

Versione di runtime 4.7.0

zOS

Nuove funzionalità

- Aggiunta una nuova proprietà YML `dataSimplifier.doubleFloatingType` per definire il formato da utilizzare per la codifica e la decodifica dei numeri a virgola mobile - I valori consentiti sono `IEEE_754` (impostazione predefinita)/(per il formato esadecimale a virgola mobile) `HFP`

- È stato aggiunto il supporto della posizione di scala decimale P sul valore numerico PICTURE

Miglioramenti

- COBOL
  - Supporto migliorato per l'istruzione INSPECT con le clausole AFTER INITIAL e BEFORE INITIAL
  - Supporto aggiornato di Numeric Edited Type con conversione ottimizzata dei byte che migliora l'efficienza complessiva della gestione dei dati
  - È stato aggiunto il supporto per FUNCTION MIN & MAX con confronto alfanumerico
  - Supporto migliorato per FUNCTION INTEGER per valori negativi
- PL/I
  - Supporto migliorato dell'istruzione PUT con FBA/VBA set di dati SYSPRINT
- JCL - ORDINA
  - Aggiunto il supporto dell'opzione OUTREC SQZ con i sottoparametri SHIFT, LEAD, TRAIL, MID, PREBLANK e PAIR
  - Attributi DCB di catalogazione migliorati per il file SORTOUT
- JCL - DSNUTILB
  - Supporto avanzato per il tipo di dati DECIMAL EXTERNAL
  - Supporto migliorato per la condizione NULLIF relativa ai valori/alla posizione di un'altra colonna
  - Supporto migliorato per le colonne BYTEA
- JCL - INFUTILB
  - Supporto migliorato per le parole chiave CURRENT [DATE/TIME/TIMESTAMP] nell'operazione UNLOAD
- JCL - Supporto migliorato per il codice di ritorno qualificato di called PROCs
- Elaborazione dati
  - Logica di confronto numerico migliorata durante l'elaborazione di campi con valori vuoti ridefiniti da tipi alfanumerici a numerici, garantendo una valutazione corretta rispetto a zero
- <noloc>Blusam</noloc>
  - Supporto migliorato per il comando START per gestire le ricerche parziali delle chiavi utilizzando segmenti della chiave primaria, offrendo funzionalità di recupero dei record più flessibili
- SQL
  - È stato aggiunto il supporto per i parametri di input passati nella clausola ON

  - Aggiunto il supporto per la funzione REPEAT
  - Supporto migliorato per l'istruzione UPDATE con un'assegnazione che coinvolge il nome della colonna
  - Gestione migliorata di SQLCODE e SQLSTATE nei recuperi di cursori su più righe
  - Aggiunto il supporto per la funzione XML per la destinazione del database DB2 DB2
- Integrazione della coda dei messaggi
  - Gestione degli errori migliorata nelle operazioni MQ GET, che fornisce un'indicazione di stato più chiara MQCC_FAILED per i corpi dei messaggi vuoti
  - Supporto migliorato per le operazioni MQ PUT gestendo l' MQRFH2 intestazione nel contesto delle transazioni XA
  - Supporto migliorato per i segreti Transform for mainframe di AWS Runtime per gestire proprietà aggiuntive per JMS MQ
  - Capacità di configurazione del pool di connessioni JMS MQ migliorata
- Misc (Vari)
  - Integrazione del sistema: è stato aggiunto il supporto per l'ambiente dello schema nelle chiamate di DFSRRC00 programma
  - Aggiunta compatibilità per le transazioni IMS quando <noloc>Blusam</noloc> è disabilitato
  - Supporto migliorato per la end-of-file condizione su un file sequenziale da allineare alla regola precedente «EOF è considerata una lettura non riuscita»

AS400

Nuove funzionalità

- È stato aggiunto il supporto per il comando RUNSQLSTM per eseguire query SQL archiviate in un file sorgente esterno

Miglioramenti

- CL
  - Aggiunto meccanismo di blocco dei file tramite i comandi ALCOBJ/DLCOBJ
  - Migliorata la funzionalità del comando CPYF per mantenere l'integrità della sequenza di registrazione durante la copia dei file
  - Comando CPYF migliorato per supportare l'output come file flat

  - Gestione avanzata dei parametri per regolare automaticamente la lunghezza dei dati durante le chiamate al programma
  - Supporto migliorato dei parametri per gestire il riferimento all'intervallo
  - Aggiunto il supporto per MONMSG CPF2105 sul comando DLTDTAARA
  - Aggiunto il supporto della libreria QTEMP per il comando CHKOBJ su DATAAREA
  - Supporto migliorato per il comando ADDLFM, miglioramento delle capacità di gestione dei membri dei file logici
  - È stato aggiunto il supporto per i nomi generici delle aree dati sul comando DLTDTAARA
  - È stato aggiunto il supporto a SBMJOB per gestire il passaggio di switch tramite il parametro SWS
  - È stata migliorata la logica di scrittura dell'area dati inserendo le chiamate di lettura e scrittura in un blocco sincronizzato
  - Gestione avanzata della concorrenza tra aree dati per prevenire scritture simultanee attraverso l'implementazione di un meccanismo di blocco granulare, eliminando gli errori durante le operazioni parallele
  - È stato aggiunto il supporto per la parola chiave *ALL nel parametro DTAARA per il comando RTVDTAARA
- Schermo
  - È stato migliorato il colore di un attributo bianco nel front-end
- RPG
  - Supporto migliorato per il funzionamento TESTB per gestire meglio diverse codifiche
  - Supporto migliorato per il funzionamento EDITC con il codice di modifica «Y» e «P»
  - Supporto migliorato per EDITCODE per i codici K, Q e Z
  - Implementa il supporto per il supporto delle ExtProc parole chiave Prototype sulla procedura locale con parametri
  - Parametri gestiti passati per valore per procedure esterne
- COBOL
  - Istruzione SORT migliorata per gestire la libreria mirata
  - Troncata una partizione quando la DAO associata viene aperta in modalità OUTPUT
  - È stato aggiunto il supporto per la parola chiave PREFIX
- DataQueue

  - Aggiunte opzioni di configurazione keystore e truststore per i dati RabbitMQ e le code di messaggi ibride, che consentono la messaggistica sicura nelle implementazioni di produzione
  - Prestazioni di riavvio del server ottimizzate grazie all'eliminazione del controllo ridondante dell'esistenza di code di dati
  - Architettura semplificata grazie alla rimozione dei componenti DataQueueRegistry DataQueueConnectionProvider
  - Eliminazione semplificata della coda di dati tramite gestione diretta senza convalida del registro
  - Processo di creazione delle code migliorato garantendo la creazione e l'associazione corrette degli exchange queue-to-exchange
- Misc (Vari)
  - Aggiunta la convalida delle dimensioni delle richieste di transazione per prevenire il sovraccarico del sistema dovuto a richieste di intervallo eccessivo
  - Migliore gestione dei blocchi sulla pulizia delle risorse lavorative
  - È stato migliorato il meccanismo di blocco DAO per impedire il blocco di un record quando il file viene aperto in modalità INPUT
  - Prestazioni ottimizzate per la mappatura dinamica delle entità
  - Migliore gestione del contesto di persistenza JPA per migliorare le prestazioni nelle operazioni DAO
  - Supporto migliorato dell'area dati aggiungendo colonne di timestamp per gestire i metadati dei record
  - Maggiore affidabilità dell'area dati implementando operazioni di lettura-scrittura sincronizzate per prevenire problemi di accesso simultanei
  - Supporto migliorato per #moveDate per supportare le conversioni DateHelper date-to-timestamp
- DDS
  - Migliore supporto delle Select/Omit condizioni combinato con operazioni sintetiche come SST o CONCAT
- SQL
  - Il SQLExecutor Builder è stato ottimizzato per gestire il recupero parziale dei record in modo più efficiente, in particolare quando si recuperano più righe con limiti di recupero superiori a uno

Motore di Transform for mainframe trasformazione AWS 4.7.0

ZoS

Miglioramenti

- COBOL
  - Aggiunto il supporto per l'istruzione CODE-SET
  - Aggiunto il supporto per l'opzione NOT END OF PAGE dall'istruzione WRITE
  - È stato aggiunto il supporto per una nuova sintassi su ALTER
  - È stato aggiunto il supporto per la sostituzione di più righe per l'istruzione COPY REPLACING
  - Migliore analisi della parola chiave CHARACTERS implicita nella definizione di clausole diverse
  - Supporto migliorato per le chiavi alternative coinvolte nella sezione input-output quando i nomi delle chiavi sono duplicati ma hanno genitori diversi
  - Supporto migliorato per la condizione IF con confronto tra campi di grandi dimensioni e spazi
  - Risoluzione statica dei riferimenti ai file logici migliorata senza distinzione tra maiuscole e minuscole
  - Trasformazione del codice migliorata per generare automaticamente campi impliciti LINAGE-COUNTER per ogni voce FD che contiene una clausola LINAGE
  - Supporto migliorato per la definizione del cluster IDCAMS con un nome racchiuso tra virgolette semplici
  - Supporto migliorato dell'istruzione PERFORM per il formato Basic, con il formato della frase TIMES e il formato che prevede la qualificazione della sezione
  - È stata abilitata la trasformazione dell'istruzione SORT con più file GIVING e aggiunta un'eccezione per tenere traccia di questo caso durante l'esecuzione
  - È stata migliorata la trasformazione di RENAMES quando specificata in un gruppo utilizzando la clausola REDEFINES
- PL/1
  - Aggiunto il supporto per l'opzione DATA o LIST per PUT STRING
  - È stato aggiunto il supporto per il programma Multiple Entry

AS400

Miglioramenti

- CL
  - Migliorata l'analisi dei parametri per QCMDEXC per poterlo eseguire tramite QCMDEXC ALCOBJ/DLCOBJ
  - È stato ElseIf aggiunto il supporto per l'istruzione su CL
  - È stato aggiunto il supporto per la funzione integrata %SWITCH per impostare i job switch.
- DDS
  - Supporto migliorato del record DSPF descritto dal programma per gestire record fittizi nelle specifiche di input
  - Migliorate le condizioni della clausola WHERE di generazione DAO nel caso di un file CREATE INDEX SQLDDS
- RPG
  - Aggiunto il supporto per l'inizializzazione ZEROES INZ (*ZEROS) su Unsigned Integer
  - Supporto migliorato del comando EXCEPT con un tipo di file Workstation e SQL Index
  - Supporto migliorato del tipo di dati interno RPG per campi numerici definiti esternamente
  - Supporto migliorato del sistema integrato %ERROR per ottenere lo stato di errore corrente dopo un'operazione su un file
  - Supporto migliorato della parola chiave COMMIT file per gestire file logici multiformato
  - SELECT/OMIT Elaborazione migliorata per file logici multiformato
  - Migliore gestione MOVEA per l'impostazione di array con spazi vuoti, lo spostamento di array fissi di Packed, Zoned e Binary su un altro array dello stesso tipo.
  - Migliore gestione degli array per Z-ADD *ZEROS e MOVEA *ZEROS
  - MOVE/MOVEL Operazioni migliorate quando si passa da combinazioni numeriche e di caratteri
  - Implementa il supporto per le ExtProc parole chiave di RPGLE Prototype sulla procedura locale con parametri
  - Migliore gestione della costante figurativa*HIVAL nelle istruzioni di assegnazione
  - È stato migliorato il supporto per la parola chiave PREFIX per gestire la sostituzione dei caratteri
- COBOL400
  - Supporto migliorato per le operazioni REWRITE e UPDATE in modalità di accesso dinamico
  - È stato aggiunto il supporto per la clausola ACCEPT Statement FOR in CBLLE
  - Supporto migliorato del sistema integrato %STATUS per ottenere lo stato corrente di un file dopo un'operazione
- Misc (Vari)

- Migliore trasformazione dei programmi che definiscono strutture di dati che condividono lo stesso nome di uno dei campi nel PF/LF specificato
- Migliore generazione di accessi agli array sulle variabili kanji

Note di rilascio 4.6.0

Data di rilascio: 24 gennaio 2025

Abbiamo testato questa versione di AWS Transform for mainframe Runtime con lo stack seguente. Potrebbero essere compatibili anche altre versioni.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.11.0
	Npm 10.9.0
	Angolare 18
Livello di servizio	Spring Boot 3.3.5
	Spring Core 6.1.14
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

Versione di runtime 4.6.0

zOS

Miglioramenti

- COBOL

  - `WRITE ADVANCING`Funzionalità avanzate con maggiore precisione per la scrittura sequenziale di righe di file, supporto di più contesti (`BEFORE>` e usi impliciti) e implementazione completa delle istruzioni `AFTER PAGE`
  - Supporto migliorato `FILLER` per i casi in cui una tabella annidata FILLER viene utilizzata come gruppo con una tabella come elemento secondario
  - Migliore accesso ai figli di genitori ambigui all'interno di un unico segmento
  - È stato aggiunto il supporto per il tipo Numeric Edited con picture='-----'
  - Migliore gestione della visualizzazione dei dati di tipo BINARY
- PL/I
  - Migliore conversione dei valori letterali binari nelle istruzioni di assegnazione
- JCL — ORDINA
  - Supporto migliorato per `OVERLAY` parametri consecutivi nella stessa istruzione `OUTFIL`
- JCL — DSNUTILB
  - Meccanismi di caricamento ottimizzati, con tempi di recupero dei dati più rapidi del 25%
  - Supporto migliorato per le transazioni XA per fonti di dati aziendali esterne
- JCL — INFUTILB
  - UNLOAD - Aggiunto il supporto del tipo di dati FLOAT8
- JCL — IDCAMS
  - Gestione ottimizzata dei codici di ritorno per i comandi `IDCAMS`
  - Aggiunto il supporto per eliminare tutte le generazioni GDG in base al nome di base GDG
  - Aggiunto il supporto per l'eliminazione di file senza parametri `NONVSAM`
- JCL — Varie
  - Gestione avanzata dei metadati di riavvio in batch per migliorare la gestione dello stato del flusso di lavoro durante la modalità di riavvio
- <noloc>Blusam</noloc>
  - Aggiunto il supporto di TTL per la cache <noloc>Blusam nelle implementazioni Ehcache</noloc> e Redis
  - <noloc>Supporto migliorato per il `DEPENDING ON` campo sulla descrizione del file COBOL per il file Blusam KSDS `FD`</noloc>
  - Maggiore sicurezza dei thread nelle operazioni di lettura di Redis <noloc>Blusam</noloc> per l'esecuzione simultanea di più processi

  - Creazione dello schema <noloc>Blusam</noloc> migliorata per una maggiore robustezza dei privilegi degli utenti del database
  - Migliore imbottitura a destra sul set di dati di input concatenato a blocchi variabili `READ`
- INDIETRO
  - È stato aggiunto il supporto per la creazione di set di dati in modalità multi-schema, inclusa una nuova colonna «Schema» per indicare l'associazione dello schema per ogni set di dati
- MFS
  - Migliore propagazione delle informazioni utente dal front-end al contesto condiviso, garantendo una corretta propagazione nel contesto JHDB
  - Aggiunto il supporto per l'intestazione informativa IBM MQ IMS sulle transazioni XA
- SQL
  - `SQLCODE`Gestione migliorata per impostare 305 durante il recupero del cursore quando tutti i valori delle colonne sono NULL
  - È stato aggiunto il supporto per la `IN` clausola che coinvolge il parametro per le condizioni `OCCURS WHERE`
  - È stato aggiunto il supporto per le istruzioni `DECLARE GLOBAL` temporanee delle tabelle
  - Supporto DB2 SQL esteso per il formato timestamp DB2 specifico di mezzanotte e 24 ore su 24 tramite conversioni dedicate in fase di esecuzione secondo un motore di database mirato
- Misc (Vari)
  - IBM930 Set di caratteri migliorato per consentire ai caratteri Unicode U+2014 e U+2015 di corrispondere a X'44x4a' in EBCDIC
  - TDQUEUE - Implementazione SQS rifattorizzata per supportare il multithreading
  - Risoluzione dei nomi del set di dati GDG migliorata per consentire al cliente di archiviare i file con lo stesso prefisso GDG (ad esempio è il file corrente ed è un file di archivio) `A.B.C.G0002V00` `A.B.C.G0001V00.1236`
  - Migliorato `SQLConverter::toPgmDate/Time/Timestamp` per allineare il calcolo della data in base al formato precedente

AS400

Nuove funzionalità

- È stato aggiunto il supporto per AS400 tabelle create dinamicamente per file flat ed entità duplicate, che consente l'accesso alle tabelle create tramite comandi CL come CRTPF, CRTDUPOBJ e CPYF
- Aggiunto un servizio per supportare l'elenco delle librerie tramite un registro che gestisce la libreria predefinita per ogni tabella

Miglioramenti

- CL
  - CLRPFM - Migliore gestione del membro quando viene chiamato il comando per la libreria QTEMP
  - SMBJOB - Supporto migliorato dei parametri per gestire argomenti creati dinamicamente `PARM`
  - CPYFRMIMPF - Aggiunto il supporto per i parametri e `TIMFMT` `ERRRCDFILE` `ERRRCDOPT`
  - CPYFRMIMPF - Supporto migliorato per i valori alfanumerici del database che contengono virgolette singole
  - CPYF - Ha perfezionato la struttura delle query di comando per file con più membri con `FROM TOMBR(*ALL)`
  - CPYF - Supporto migliorato per la gestione dei parametri per `FMTOPT` `MAP DROP`
  - CPYTOIMPF - Supporto migliorato dei parametri per la gestione della tabella `FROMFILE` `MEMBER`
  - RTVUSRPRF - Aggiunto il supporto per i parametri `RTNUSRPRF`
  - DSPDBR - Modifica il comando per adattarlo al comportamento precedente previsto, consistente nella stampa delle informazioni sulle viste esistenti in una tabella, nonché sulla libreria e sul membro di cui fanno parte
  - DSPFD - Supporto migliorato dei parametri `FILE`
  - DSPFD - Supporto migliorato dell'`TYPE MBR`output dei parametri per includere valori aggiuntivi: mbfile, mblib, mbfcdt, mfccn
- Schermo
  - Migliore priorità di posizione del cursore per `DSPATR(PC)`
  - È stata migliorata la convalida dei campi dei record dei sottofile ignorando la convalida front-end dei campi «protetti»
  - Supporto migliorato per l'inizializzazione dei record nella workstation con più campi di matrice che condividono i nomi dei componenti

  - Supporto migliorato per gli indicatori di risposta nelle `DSPF` parole chiave (`SFLMSGSFLMSGID`, `CHANGE` e tasti di comando)
- RPG
  - Supporto migliorato del ciclo di programma per una migliore gestione dei campi letti dai file primary/secondary
  - Aggiunto il supporto per Split Control Field per la lettura primary/secondary dei file
  - Metodo `%SUBST` integrato migliorato per gestire i campi a doppio byte nelle istruzioni di confronto
  - Supporto migliorato dell'indicatore ZERO per il funzionamento di MVR
- DDS
  - È stato aggiunto il supporto di file logici multiformato con formato di registrazione che si riferiscono allo stesso record fisico
- DataQueue
  - Migliore gestione delle interruzioni di lavoro per i lavori in attesa di ricevere messaggi in coda dai dati grazie alla possibilità di ripulire l'utente durante le interruzioni
  - Migrato da RabbitMQ a Spring-AMQP per una migliore gestione dei canali e il ridimensionamento dei thread
- Misc (Vari)
  - SQLExecutorBuilder migliorato per supportare query con più spazi bianchi e parentesi aperte senza spazi iniziali
  - Supporto DAO migliorato per gestire correttamente il posizionamento del cursore mentre si cambia la direzione di lettura
  - Inizializzazione delle chiavi raffinata dopo le operazioni di recupero ed eliminazione per garantire la corretta rimozione dei record correlati prima di inserire i record aggiornati
  - Codice generato da DAO mapper ottimizzato per migliorare le prestazioni di esecuzione temporale

AWS Transform per mainframe Transformation Engine 4.6.0

zOS

Miglioramenti

- COBOL
  - Analisi migliorata della `RESERVE` clausola con valori letterali opzionali `AREA/AREAS`

  - Supporto COBOL migliorato con `DATA DIVISION` dichiarazione opzionale, che supporta casi di test semplificati
  - Paragrafo con nomi speciali migliorato aggiungendo il supporto per `ALPHABETSYMBOLIC`, e `CLASS` clausole, switch e variabili `FORMFEED`
  - È stato aggiunto il supporto `SYSIN` come nome mnemonico nelle istruzioni `ACCEPT`
  - Supporto avanzato delle `PICTURE` clausole per i simboli «$», «0", «CR», «DB» nei calcoli delle dimensioni logiche `PIC`
  - Trasformazione delle `USE` istruzioni migliorata per più scenari di file
  - Trasformazione avanzata delle `ALTER` dichiarazioni per più modifiche
  - È stato aggiunto il supporto per le costanti figurative nella clausola `ZERO` `HIGH-VALUE` `LOW-VALUES` `delimited by`
- SQL
  - Migliore trasformazione del valore predefinito per il target PostgreSQL per gestire le virgolette attorno al valore predefinito `CURRENT_TIMESTAMP`
  - Gestisci la `WITH CHECK OPTION` clausola delle viste SQL

AS400

Miglioramenti

- DDS
  - Supporto migliorato per file logici multiformato che fanno riferimento più volte allo stesso record fisico
- RPG
  - Funzionamento avanzato `MOVE` per gestire `MOVEL` meglio gli zeri di imbottitura
  - Gestione migliorata delle chiamate di funzioni annidate nelle valutazioni e nelle condizioni
- COBOL400
  - È stato aggiunto il supporto per la trasformazione della `IN` parola chiave in istruzioni `SELECT`
  - Supporto migliorato per i punti mancanti nelle voci di descrizione dei dati, in linea con l'ultima versione COBOL, in cui si presuppone che i punti siano mancanti
  - Posizionamento migliorato del cursore sulle operazioni `REWRITE`
  - Supporto migliorato per l'`START`istruzione di blocco del record nella posizione corrente del file

  - Supporto migliorato per la direttiva del compilatore `COPY DDS` per generare tutta la struttura input/output dei dati
- Misc (Vari)
  - StateMachines - Trasformazione migliorata per migliorare la dichiarazione degli stati compositi in linea con il paradigma stateless4j
  - Migliore sanificazione per i file LF contenenti caratteri speciali
  - Supporto migliorato del formato figurativo `*ALL` con valori esadecimali
  - Supporto `MOVE` operativo migliorato per la conversione implicita da tipi numerici a tipi di caratteri
  - Generazione ottimizzata di report bean per l'ordinamento in base al nome della stampante associata, evitando nomi duplicati o in conflitto
  - Supporto migliorato per le parole chiave `EXTFILE` combinato con `USROPN` la gestione del valore letterale e del formato `libname/filename`

Note di rilascio 4.5.0

Data di rilascio: 20 dicembre 2024

Questa versione di AWS Transform for mainframe Runtime e AWS Transform for mainframe Transformation Engines include le seguenti funzionalità chiave.

- Supporto JCL: ora è possibile generare ed eseguire script JCL al volo all'interno del contesto di runtime. Questa funzionalità aggiunge flessibilità e automazione nell'elaborazione dei lavori in batch. Abbiamo aggiornato il supporto per le utilità JCL in fase di esecuzione, con una serie di miglioramenti a SORT, ICETOOL, INFUTILB e IDCAMS (vedi i dettagli nelle sezioni seguenti). Questi miglioramenti offrono funzionalità di elaborazione dei dati più solide ed efficienti.
- Binding Directories and Activation Groups Support for AS/400 Modernized Applications — Binding Directories migliora l'organizzazione del sistema gestendo i riferimenti alle procedure esportati, mentre i Gruppi di Attivazione semplificano la gestione del contesto di esecuzione. Queste funzionalità migliorano la precisione e l'affidabilità, una solida gestione delle risorse e le interazioni di sistema ottimizzate. Il risultato è un sistema più resiliente, organizzato ed efficiente per applicazioni modernizzate AS400 .
- Aggiornamenti delle dipendenze: — Aggiornamento di tutti i framework di frontend (BAC/JAC e applicazioni modernizzate) alle versioni con supporto a lungo termine (LTS). L'aggiornamento di Angular dalla v17 alla v18 introduce un nuovo modello di reattività e una gestione degli stati

semplificata, riducendo la complessità e migliorando la manutenzione delle applicazioni per gli sviluppatori. Node.JS è stato inoltre aggiornato dalla v20 alla v22.

Abbiamo testato questa versione di AWS Transform for mainframe Runtime con lo stack seguente. Potrebbero essere compatibili anche altre versioni.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 22.11.0
	Npm 10.9.0
	Angolare 18
Livello di servizio	Spring Boot 3.3.5
	Spring Core 6.1.14
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

Versione di runtime 4.5.0

ZoS

Nuove funzionalità

- JCL — Aggiunta la possibilità di richiamare un processo in batch da programmi online. Abbiamo aggiunto un servizio per gestire gli script JCL memorizzati in un file dedicato TDQueue quando un programma modernizzato lo genera al volo. Questo servizio consente di ricostruire il messaggio JCL, rifattorizza questo messaggio in uno script groovy ed esegue questo script groovy.
- ADABAS — Aggiunto il supporto per il programma ADABAS. Con questo supporto, il runtime emula i comandi ADABAS per l'accesso al database (disponibile solo per Oracle).

Miglioramenti

- COBOL
  - Supporto migliorato dell'istruzione DISPLAY che sfrutta l'opzione NO ADVANCING
  - Maggiore precisione nella gestione dei segni valutari che consente all'utente di beneficiare di una struttura COBOL trasformata più accurata
  - Supporto migliorato per l'assegnazione di valori quando si sposta un campo non firmato in un campo firmato e viceversa
  - Supporto migliorato per la dimensione dei blocchi per file GDG e file concatenati
- CICS
  - Aggiunto il supporto per OpenStatus e EnableStatus dei set di dati <noloc>Blusam</noloc>
  - È stato aggiunto il supporto per il comando `SET DATASET`
- JCL — ORDINA
  - Migliore gestione delle dimensioni dei record del set di dati
  - Supporto migliorato per l'`OUTFIL`istruzione per produrre file di output contenenti solo i record dei file di input in base ai valori specificati in STARTREC e nelle opzioni `ENDREC`
  - Supporto migliorato per le istruzioni `OVERLAY`
  - Supporto migliorato per l'`OUTREC`istruzione per gestire una variante dell'`EDIT`opzione. Ora supportiamo `EDIT(...)` in aggiunta a `EDIT=(...)`
  - È stato aggiunto il supporto per il pattern (p, m, f, OPERATOR, p2, m2, f2) nelle operazioni aritmetiche
  - È possibile utilizzare la clausola `DUMMY` file del `SORT` programma di un JCL per gestire file di input vuoti e trarre vantaggio dalla generazione di file vuoti
- JCL — ICETOOL
  - Supporto migliorato per la `SORT FIELDS=COPY` dichiarazione tramite il programma `SORT`
- JCL — INFUTILB
  - Supporto migliorato per il calcolo delle dimensioni dei record se non è specificato nel JCL e la proprietà DFSIGDCB è disabilitata
  - È stata migliorata la clausola UNLOAD with INTO per DECIMAL aggiornando la precisione e la scala in base ai campi della clausola into
  - Metodo di formattazione migliorato in VarcharFormatter
  - Supporto migliorato con una nuova opzione configurabile che consente agli utenti di controllare il modo in cui i campi VARCHAR vengono gestiti durante lo scaricamento dei dati per quanto

riguarda il comportamento di riempimento, garantendo flessibilità e precisione nei processi di estrazione dei dati.

- JCL — IDCAMS
  - Eliminazione migliorata per i file con suffisso wildcard e nome definiti direttamente tra parentesi o tra virgolette semplici
  - Maggiore precisione per sfruttare il codice di ritorno MAXCC
- JCL — IKJEFT01 - Aggiunto il flag di funzionalità `systin.encoding` (default `=ASCII`) per supportare una codifica specifica per il set di dati di file SYSTSIN
- JCL — Supporto migliorato per la proprietà BDW per un file di output generato in una fase JCL e i passaggi successivi utilizzano lo stesso file system di input e `DISP=PASS`
- MF
  - Supporto migliorato per l'intestazione a 2 byte per il file Record Sequential
  - Migliore gestione dei codici di ritorno per il comando DELETE
  - Linea di avanzamento della scrittura migliorata per il file Record Sequential
- Redis
  - Inizializzazione del modello Redis migliorata per checkpoint JCL e Jics TSQueues
  - Accessibilità e leggibilità delle informazioni di blocco dei record del set di dati Redis migliorate
- SQL
  - Migliore analisi di FOREIGN KEY con clausola REFERENCES
  - Ha fornito una funzionalità di memorizzazione nella cache estensibile per archiviare i tipi grafici legacy originali nel database, migliorando la tracciabilità dei dati e facilitando il calcolo grafico
  - Supporto avanzato per l'analisi del pattern CASE WHEN delle query SQL tra le utilità di runtime
  - La funzione AWS Transform for mainframe integrata gwdecimal di SQL Postgres migliorata su cui si basa il runtime per adattarsi alla funzione integrata DECIMAL. DB2
- Misc (Vari)
  - Supporto NumericEditedType migliorato per l'utilizzo dell'operando SIGN
  - Migliore generazione della configurazione dell'origine dati primaria SpringBootLauncher nell'applicazione modernizzata
  - Maggiore flessibilità per separare i log delle applicazioni dal percorso relativo al lavoro chiamato.
  - Supporto migliorato per il valore vuoto nel confronto dei campi da NumberUtils
- FILE — Supporto migliorato dei set di dati a blocchi variabili nei file sottostanti
- MQ — Gestione migliorata delle connessioni MQ per ambienti ad alta disponibilità

- Migliore compatibilità con MQ Queue aggiungendo il supporto per client non JMS per migliorare la codifica e la gestione dei set di caratteri
- Supporto migliorato per i caratteri di controllo ANSI per il file Ebcdic

AS400

Nuove funzionalità

- È stato aggiunto il supporto per i dati esportati all'interno dei programmi associati
- Aggiunto il supporto specifico ILE per la divisione per zero

Miglioramenti

- COBOL400
  - Supporto migliorato di EOF in File Status
  - Aumenta il supporto di precisione dell'istruzione Cobol START per supportare la parola chiave EQUAL nella clausola KEY IS
- CL
  - Aggiunto il supporto per il comando UPDENVPARM
  - CRTPF - Aggiunto il supporto per la tabella a cui si accede con una partizione
  - RCVF - Supporto migliorato dei file logici con override
  - FTP - Supporto migliorato per I/O i file logici con OVRDBF e registro OUTPUT migliorato e aggiunto il supporto per i file nella directory di lavoro I/O
  - CPYFRMIMPF - Aggiunto il supporto per i parametri, `ERRRCDFILE` `TIMFMT` `ERRRCDOPT`
  - CPYF - Creazione di partizioni QTEMP migliorata
  - CPYF - Aggiunto messaggio di monitoraggio quando il file*FROM è vuoto
  - OVRPRTF - Aggiunto il supporto per nuovi parametri:`PAGESIZE`,,,,,, `OUTQ` `DEV` `LIP` `CPI` `OVRFLOW` `LVLCHK` `FORMTYPE` `HOLD`
  - Maggiore precisione quando si utilizza il `FMTOPT` parametro con `MAP` e `DROP` le opzioni del `CPYF` comando per consentire la copia dei dati da un file di origine con colonne aggiuntive a un file di destinazione
  - Maggiore precisione nella gestione della mappatura dei pattern jolly del percorso del file system nel comando `RMVLNK`

  - Il comando RMVM (Remove Virtual Machine) è stato migliorato per gestire le tabelle DROP delle partizioni garantendo la pulizia completa delle risorse correlate.
  - OPNQRYF - Supporto migliorato del parametro *FILE per il comando
  - Implementata CPF0000 la gestione per comprendere tutti i messaggi CPFx
  - CHGDTAARA - Aggiunto il supporto per la parola chiave *ALL per modificare l'intera area dati
- Schermo
  - tables/subfile Visualizzazione migliorata aumentando la precisione dello scorrimento e del cursore position/priority
  - CHECK(RB)Funzionalità migliorata per campi non numerici CHECK(RZ) e non firmati
  - Supporto migliorato della funzionalità della schermata di aiuto per le parole chiave HLPARA
- RPG
  - Supporto migliorato del sistema integrato %SubDt
  - Supporto migliorato per le procedure che utilizzano una struttura di dati locale descritta esternamente
  - È stato aggiunto il supporto per il parametro opzionale del codice di errore e QMHSNDPM QMHRMVPM QMHRCVPM
  - Supporto migliorato del metodo %SUBST integrato per gestire meglio i campi a doppio byte.
  - È stato aggiunto il supporto per %TLOOKUP integrato e le sue varianti (%TLOOKUPGE, %TLOOKUPGT, %TLOOKUPLE, %TLOOKUPLT)
- Area dati
  - Supporto migliorato per l'operazione OUT quando il fattore 1 è vuoto
  - Letture simultanee migliorate sulla stessa area dati
  - Aggiunta una variabile di configurazione blu4iv.dtaara.library.disable per disabilitare le librerie per l'area dati
  - Supporto esteso per sfruttare le librerie denominate tramite operazioni di area dati che consentono all'utente di strutturare la posizione dell'area dati come desidera.
- DataQueue
  - Utilizzo migliorato del canale RabbitMQ
  - RabbitMQ Consumer è stato migliorato, in modo che tenti di annullare il cliente una sola volta
  - È stato migliorato il recupero della coda di dati da RabbitMQ tentando BasicGet solo quando il tempo di attesa è 0
- Misc (Vari)

- Spazio utente: comportamento migliorato quando più processi tentano di recuperare lo stesso spazio utente contemporaneamente
- Supporto migliorato per l'eliminazione senza impegno dei record sotto il controllo dell'impegno
- Entità: supporto migliorato per le omissioni consecutive poiché OMIT ha un significato implicito `AND`
- È stato aggiunto il supporto per camel case nei setter Entities Mappers per gestire i nomi personalizzati definiti tramite un ulteriore refactoring
- Migliore propagazione delle informazioni utente dalle AS400 transazioni ambientali attraverso l'intera applicazione.
- Maggiore precisione al termine di un lavoro programmato da Quartz in caso di interruzione.
- Supporto migliorato per Commitment Control per renderlo applicabile al programma

AWS Transform per mainframe Transformation Engine 4.5.0

ZoS

Miglioramenti

- JCL - Generazione groovy migliorata per il set di dati KSDS basata sull'analisi LISTCAT
- COBOL
  - Analisi migliorata dell'`COPY-REPLACING`istruzione per gestire la sostituzione del sottocampo qualificato quando è presente ambiguità per questo nome di sottocampo
  - Supporto migliorato per l'istruzione defined in `SYSOUT SPECIAL-NAMES`
  - Supporto migliorato degli ZEROES figurativi nell'istruzione `ADD n TO ZERO`
  - Supporto migliorato per l'`REPLACE`istruzione per gestire problemi su più righe appiattendo le chiavi e i blocchi di testo su più righe
  - Supporto migliorato per le operazioni aritmetiche con clausola ADD/SUBTRACT/MULTIPLY/DIVIDE `GIVING`
  - È stato avviato il supporto all'analisi di REPORT SECTION e delle relative azioni (INITIATE, TERMINATE, GENERATE report)
- Varie - Migliora la generazione e la robustezza dei bollettini meteorologici

AS400

Miglioramenti

- DDS
  - Supporto migliorato della lunghezza implicita di tipo DATE
  - Supporto migliorato del stop-zero-suppression carattere nella parola chiave EDITWORD
  - Supporto migliorato per il nome di colonna DESC in quanto è una parola riservata in DB
- RPG
  - Supporto migliorato per %TIME integrato
  - Migliore generazione di istruzioni EVALR per gestire l'assegnazione da un valore di stringa a una variabile di lunghezza inferiore con una migliore regolazione a destra
  - Analisi SQL migliorata dell'impostazione delle opzioni
  - Supporto migliorato per l'inizializzazione PSDS nei programmi NOMAIN RPGLE
  - Supporto migliorato della parola chiave LIKE per definire un campo numerico DDS come Packed, indipendentemente dalla descrizione esterna
  - Migliore disinfezione dei nomi di file sostituendo «$» con «DL»
  - Supporto migliorato del %SUBST integrato per gestire i valori a doppio byte
- COBOL400
  - Schermata: supporto migliorato del record DSPF per le operazioni I/O
- CL
  - Migliore ridenominazione dei nomi delle variabili riservate
  - Migliore supporto delle Select/Omits condizioni per la gestione di file in più formati
- Misc (Vari)
  - Riduzione delle entità duplicate relative alle operazioni sui file (EOF, FOUND, EQUAL)
  - Migliore generazione di file JRXML per QPRINT, una stampante standard su AS/400. Quando viene utilizzato, il file JSON creato non conterrà alcun riferimento al programma o al file. Viene generato un solo file JRXML (Qprint-qprint.jrxml)
  - È stata migliorata la visualizzazione di informazioni aggiuntive sui messaggi per i componenti che visualizzano i messaggi dalla coda del programma

Note di rilascio 4.4.0

Data di rilascio: 13 novembre 2024

Questa versione di AWS Transform for mainframe Runtime and Transformation Engines si concentra sull'aggiornamento delle dipendenze critiche e delle tecnologie supportate, aumentando al contempo le prestazioni in molteplici funzionalità. Alcune funzionalità e modifiche chiave di questa versione includono:

- Aggiornamenti delle dipendenze: le applicazioni console (BAC e JAC) e le applicazioni modernizzate sono ora in esecuzione su Bootstrap 5. Il AWS Transform for mainframe Runtime è ora basato sul framework Spring Boot 3.3.5.
- Prestazioni: sono state migliorate le prestazioni di esecuzione delle macchine a stati (fino a 10 volte più veloci), grazie a una nuova implementazione che supera il peggioramento delle prestazioni dopo l'aggiornamento della libreria Spring State Machine dalla versione 2.5.1 alla 4.0.0. Questo aggiornamento non era facoltativo in quanto la versione 2.5.1 non era più mantenuta e contiene Critical and High. CVEs Include un'implementazione di una macchina a stati di runtime sulla piattaforma in una nuova libreria, con un'implementazione di macchine a stati leggera ed efficiente, priva di CVE e con prestazioni complessive migliori.
- Semplificazione dell'accesso al database: è stata completata una revisione significativa dei componenti utilizzati per accedere al database, tra cui DAOs entità JPA, entità DDS e Mapper. DataSimplifier Questa riprogettazione è stata determinata dalla necessità di fornire un supporto migliore per la funzionalità OVRDBF (Override Database File) comune nei progetti. AS400 Consente di gestire più casi con un'architettura semplificata per il codice generato.

Abbiamo testato questa versione di AWS Transform for mainframe Runtime con lo stack seguente. Potrebbero essere compatibili anche altre versioni dei componenti.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 18.18
	Npm 9.8
	Angolare 17

Livello di servizio	Spring Boot 3.3.5
	Spring Core 6.1.14
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

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

Versione di runtime 4.4.0

ZoS

Nuove funzionalità

- COBOL - Aggiunto il supporto per l'istruzione JSON GENERATE
- COBOL - Aggiunto il supporto per i blocchi di controllo
- MF - Aggiunto il supporto per la direttiva del compilatore FCDREG
- <noloc>Blusam</noloc> - Funzionalità aggiunta dei set di file VSAM con un'implementazione basata sullo schema del database - Supporta solo PostgressQL
- <noloc>Blusam</noloc> <noloc>- Aggiunto il supporto per la gestione del TTL (Time to live) per gli elementi di dati memorizzati nella cache di Blusam (motore di cache Redis)</noloc>
- JCL - IDCAMS - Aggiunta una nuova proprietà per forzare il set di caratteri utilizzato per decodificare la scheda SYSIN `idcams.encoding.forced`
- JICS - Estesa la `jics.db.dataScriptLocation` proprietà da `application-main.yml` per accettare un elenco di percorsi di file e cartelle. L'ordine dell'elenco è importante. Il primo file SQL viene eseguito per primo e così via. Quando una cartella viene eseguita, gli script SQL in essa contenuti vengono eseguiti senza un ordine definito.
- È stato aggiunto il supporto dell'utilità CEE3 ABD

Miglioramenti

- <noloc>Blusam</noloc> - Miglioramento del tempo di caricamento e dell'ingombro della memoria dai grandi set di dati legacy a <noloc>Blusam per</noloc> i clienti che utilizzano il motore PostgreSQL (abbiamo osservato un aumento fino a 8 volte della velocità di caricamento per set di dati di grandi dimensioni)
- <noloc>Blusam</noloc> - API exportDataSet ToS3 migliorata con supporto per credenziali
- <noloc>Blusam</noloc> - Migliorato il caricamento di file LISTCAT per la creazione di set di dati
- <noloc>Blusam</noloc> - Supporto migliorato per Dynamic READ utilizzando una chiave esplicita
- <noloc>Blusam - Migliorata la logica del meccanismo write-behind</noloc>
- JCL - Supporto JES migliorato per migliorare il blocco dei file in esecuzione parallela
- JCL - Aggiunto il supporto per la dichiarazione `INCLUDE MEMBER`
- JCL - DNSUTILB - Supporto migliorato per la chiave duplicata per gestire casi speciali in cui la chiave primaria contiene spazi
- JCL - DSNUTILB - Migliorato per ottimizzare le prestazioni durante il caricamento di dati GRAFICI LoadTask
- JCL - INFUTILB - Aggiunto il supporto per quando non è definito `fetchsize` `chunksize`
- JCL - INFUTILB - Supporto migliorato per le query che restituiscono set di risultati vuoti
- JCL - INFUTILB - Migliore robustezza durante l'elaborazione dei dati in CHUNK
- JCL - INFUTILB - Supporto migliorato per lo scaricamento con campo nullable
- JCL - INFUTILB - Supporto migliorato per il tipo numerico
- JCL - INFUTILB - Scarico migliorato per Nullable Field
- JCL - SORT - Supporto migliorato per la sintassi OUTREC
- JCL - SORT - Migliore analisi della dichiarazione DATE1
- JCL - SORT - Supporto migliorato della clausola INREC PARSE con RDW
- JCL - SORT - Formattazione dei campi migliorata utilizzando maschere di modifica
- JCL - SORT - Supporto migliorato di '' SubString in OUTREC
- JCL - SORT - Supporto migliorato per la SCHEDA compatibile con MF
- JCL - UNLOAD - Supporto migliorato delle dimensioni del campo con Postgresql
- JCL - IDCAMS - Prestazioni migliorate per il caricamento di file del set di dati VSAM grazie all'introduzione della modalità bulk
- PL/1 - Migliora il supporto per la formattazione per evitare discrepanze di scala NumericEditedType

- IMS - Supporto migliorato per la colonna destra del database IMS in NodeSorter
- CICS - Comando migliorato `RECEIVE MAP` con `SET` e no `INTO`
- BMS - Supporto migliorato del valore iniziale del campo
- SQL: DateTimeFormat analisi migliorata per il pattern `ddMMMyy`
- COBOL - Supporto migliorato per il NumericEditedType valore quando si ottiene il valore non viene considerato il punto decimale
- Supporto migliorato per la lettura di campi a lunghezza variabile in file sequenziali di righe
- Supporto migliorato per l'ereditarietà delle dimensioni dei record dal catalogo dei set di dati per i file GDG
- Supporto migliorato per la stampa di report grazie alla possibilità di personalizzare le linee di avanzamento
- Migliore inizializzazione dei dati di registrazione per i file Variable Block (VB)

GS21

Nuove funzionalità

- Schermata: è stato aggiunto il supporto per i file PSAM
- Screen - È stato aggiunto il supporto per ATTR2
- È stato aggiunto il supporto per l'ecosistema AIM (Advanced Information Manager).
- Aggiunto il supporto PED in AIM

Miglioramenti

- BitUtils Firme migliorate da gestire RangeReference
- Supporto migliorato per DummyFileConfiguration aggiungere gli attributi RecordSize/rdw/bdw/blksize/blkszlim
- Supporto migliorato per l'istruzione VPOINT per gestire il caso di un record non trovato
- Maggiore robustezza durante l'accesso all'array di record byte
- Mappatura dei caratteri del set di caratteri JEF migliorata
- Supporto migliorato per la gestione di array e condizioni nella mappatura JDBC
- Supporto migliorato per le richieste SQL nelle diverse istruzioni NDB, migliore gestione delle variazioni della sintassi SQL utilizzando costanti per ogni parte di una query SQL.

- Supporto migliorato per far sì che GS21 PackedType last nibble sia C, D o F per la convalida numerica
- Screen - Supporto migliorato per ACSAPI e per SPA e ENTER DefaultPsamController
- Schermata - Supporto migliorato per i verbi ACSAPI e NDB

AS400

Nuove funzionalità

- È stato aggiunto il supporto per i file di database in formato multi-record
- Riprogettato il framework di accesso al AS400 database
  - Funzionalità avanzate relative all'override dei file
  - Rimossi i componenti obsoleti e riduzione della complessità
  - Ha semplificato il codice generato dai programmi legacy
  - Componente DAOCycle Manager integrato nel plugin Blu4iv, che ci consente di sfruttare le funzionalità AS400 specifiche del nostro runtime personalizzato.
- JOB - Supporto migliorato per la gestione dei lavori (Quartz) per aggiungere la possibilità di interrompere una serie di lavori. job/group È stato aggiunto un endpoint dell'API REST per interrompere un lavoro con l'ID di esecuzione specificato (unico per ogni lavoro poiché è una chiave primaria). In caso di interruzione riuscita, il runtime aggiorna lo stato del lavoro su «INTERROTTO».
- È stato aggiunto il supporto per il programma di utilità CEERAN0
- Aggiunto il supporto per la modalità passiva. Aggiunto lo YAML `configuration gapwalk-application.cl:ftpservice:passive` per abilitare la modalità passiva
- Aggiunta funzionalità per creare sessioni QTEMP e ritardare la pulizia di QTEMP
- Aggiunto il supporto per la funzionalità di compilazione BNDDIR per definire dipendenze esplicite tra i programmi
- Aggiunto il supporto per il meccanismo dei gruppi di attivazione

Miglioramenti

- CL - Comando RMVMSG migliorato nella coda dei messaggi del programma per gestire la parola chiave*PREV
- CL - Supporto migliorato per le sostituzioni in OPNQRYF

- CL - Aggiunto il supporto per i parametri MSGLEN e SECLVLLEN per il comando RTVMSG
- CL - Supporto migliorato per CRTDUPOBJ per la gestione dei casi in cui NEWOBJ non viene passato e aggiunto il supporto per i nomi di tabella generici
- CL - Supporto migliorato dell'FTP per la gestione dei parametri GET, RMTSYS e BINARY
- CL - Migliorate le prestazioni delle query CLRPFM e aggiunta un'opzione per utilizzare TRUNCATE anziché DELETE
- CL - SBMJOB migliorato per gestire correttamente il parametro USER per utilizzarlo come USER quando viene inviato un lavoro
- CL - Supporto migliorato dei comandi DLTOVR per gestire il caso di*ALL
- Area dati: supporto migliorato per Blu4 DataArea aggiungendo la registrazione per la gestione delle eccezioni
- Area dati: supporto migliorato per Blu4 per DataArea recuperare una nuova DataAreaDao istanza per ogni thread
- Area dati: blocchi dell'area dati migliorati, evitando blocchi a livello di record e utilizzando invece il meccanismo di blocco appena implementato
- Area dati - L'operazione di scrittura dell'area dati ora continua con l'esecuzione quando non viene acquisito un blocco e viene fornito un indicatore di errore
- Report - Supporto migliorato per la path/naming convenzione di output dei report per i report stampati. Ha consentito ai clienti di personalizzare anche il percorso di output del report e il nome. Il cliente può specificare il proprio percorso e la propria convenzione di denominazione senza influire su nessun altro progetto.
- JOB - Supporto migliorato per la gestione dei lavori (Quartz) per aggiornare lo stato del lavoro in caso di interruzione anomala del lavoro, ad esempio: «Arresto» o «arresto anomalo» di Tomcat
- Schermata - Migliore gestione del valore numerico sul campo con modifica della parola con meno
- Screen - Popup di rendering migliorato con solo titleColorTop
- Schermata: supporto migliorato per il recupero delle informazioni di aiuto per gestire il caso in cui non venga trovato un elemento di aiuto generale
- Schermata: è stata migliorata la visualizzazione della schermata «informazioni aggiuntive» quando si preme F1 sulla riga del messaggio del sottofile
- Schermata: visualizzazione migliorata dei piè di pagina delle righe dei messaggi per SFLMSG
- Schermo: front-end migliorato per rimuovere un record nella sua interezza quando un nuovo record si sovrappone

- Accodamento: recupero dei messaggi RabbitMQ migliorato per consumare meno risorse
- Accodamento: implementazione migliorata di RabbitMQ Data Queue per recuperare solo un messaggio alla volta.
- SQL - SQLExecutor Migliore gestione da parte di Builder di SQLCODE per le query dinamiche sulle tabelle CREATE e DROP
- SQL - Supporto migliorato di OVRDBF sulle query
- SQL - SQLExecutor Builder migliorato in modo che le sostituzioni OVRDBF vengano applicate alle istruzioni preparate
- RPG - Supporto migliorato per le specifiche di input e output dei file Disk descritti dal Programma
- RPG - Supporto migliorato per la lettura dei file primari e secondari con l'indicatore MR (Matching Records). L'ordine di recupero di un ciclo DAO con campi di confronto è stato migliorato.
- RPG - Supporto migliorato per i file primari e secondari. Miglioramento dell'aggiornamento dei file primari e dell'output. File secondari. Generazione update/write del codice.
- RPG - Aggiunto il supporto per l'istruzione RETURN in formato libero
- RPG - Migliore trasformazione e gestione in fase di esecuzione delle assegnazioni numeriche decimali,
- RPG - Migliore generazione di variabili binarie
- RPG - Supporto migliorato per EDITC
- RPG - Migliore gestione dell'area dati locale
- Supporto migliorato dei campi DDS condivisi da più tipi di dispositivi (DISK, WORKSATION, PRINTER)
- Migliore gestione degli override in modo che le sostituzioni attivate non abbiano più effetto PFs LFs
- Blu4 migliorato per ivWebController non reimpostare il nome utente e l'ID utente ai valori predefiniti
- Migliore regolazione dell'indice durante le letture dei record quando la direzione di lettura cambia
- Posizionamento del cursore migliorato sulle letture dei record dopo le operazioni update/delete
- Supporto migliorato per la lettura su un DAO multientità quando la direzione di lettura cambia
- Supporto migliorato per gli spazi utente per evitare che l'istanza venga riutilizzata da tutti i thread anziché che ogni thread abbia la propria istanza
- Supporto migliorato per l'accesso simultaneo multi-threading alla lettura dei record
- È stata migliorata la memorizzazione della configurazione username/userid in tramite YML SharedContext

- Rilascio migliorato dei record bloccati con valori aggiornati
- È stato aggiunto il supporto per il comportamento specifico del compilatore OPM per l'istruzione NEXT SENTENCE

Funzionalità trasversali

Nuove funzionalità

- La nuova proprietà metadata.ini aggiunta `legacy.compilerto` specifica il compilatore legacy degli artefatti da trasformare. Il supporto di alcune istruzioni COBOL, come NEXT SENTENCE, è diverso a seconda del valore impostato.
  - «ZOS» per un sistema z/OS legacy.
  - «ILE» o «OPM» per AS400 il sistema. Impostazione predefinita = «ILE» quando `legacy.system` = «as400"

Miglioramenti

- Front-End: riprogettato i componenti del campo dello schermo per ampliare la gamma di tipi di campo supportati. Questo miglioramento consente al runtime di soddisfare una più ampia varietà di requisiti di input e dati degli utenti. AS400
- Metodo migliorato `isValid()` per l'attivazione di byte di segno separati ZonedType
- Supporto migliorato `StringConcatenationBuilder::withPointer` per la concatenazione che coinvolge CRLF
- Supporto migliorato per la codifica specifica a doppio byte per renderli thread-safe
- Prestazioni migliorate delle macchine a stati grazie all'integrazione di un nuovo framework
- Algoritmo migliorato per l'ottimizzazione delle assegnazioni per prevenire riscritture impreviste

AWS Transform per mainframe Transformation Engine 4.4.0

ZoS

Miglioramenti

- LISTCAT - Parser migliorato per evitare inserimenti duplicati
- LISTCAT - Supporto migliorato dell'ESDS al file system in JCL/Groovy

- CICS - Supporto migliorato per LENGTH OF per le istruzioni CICS

AS400

Miglioramenti

- Miglioramento della generazione di record DDS
  - È stato migliorato il supporto del record DDS per generare entità che corrispondono alla struttura dei record DDS
  - Ha fornito supporto per campi condivisi e funzioni di mappatura che si adattano meglio alla versione precedente
  - È stata migliorata la gestione dei file descritti esternamente e dei file descritti dal programma
- RPG - Rilevamento RPG migliorato per il modulo con solo formato libero
- RPG - Supporto migliorato per l'istruzione COPY che ignora la parola chiave `*LIBL/` come prefisso per individuare un quaderno dell'applicazione
- RPG - PF - Supporto migliorato per le specifiche di input con i record fisici di pfile
- RPG - Aggiunto il supporto alla dichiarazione On-Exit
- RPG - Supporto migliorato per le parole chiave LikeRec
- RPG - Mappatura migliorata dei campi DSPF rinominati
- CL - Migliore risoluzione dei nomi dei campi
- COBOL - Supporto migliorato per la conversione da esadecimale a carattere
- Supporto migliorato per la generazione di tipi decimali
- Supporto migliorato del messaggio FIXME per il codice legacy non supportato (visualizza l'intera riga legacy)
- Prestazioni migliorate su AWS Transformation Engine (fase di AS400 analisi)
- Supporto migliorato di Keyword LikeRec per allinearlo alle specifiche dei file
- Supporto migliorato della funzione integrata %Diff
- È stato aggiunto il supporto per caratteri speciali (simbolo di valuta) sull'etichetta DSPF

Note di rilascio 4.3.0

Data di rilascio: 16 settembre 2024

Questa versione di AWS Transform for mainframe Runtime and Modernization Tools si concentra sull'estensione delle capacità e della copertura per modernizzare le funzionalità del mainframe. Alcune funzionalità e modifiche chiave di questa versione includono:

- CICS: supporto aggiuntivo per lo scambio di dati dai terminali ed esecuzione di transazioni con i dati in entrata supportando il comando SEND MAP con Map Reference.
- JCL: nuova funzionalità che consente di riavviare l'esecuzione più recente di un processo batch da un JCL/PROC passaggio precedentemente fallito o di attivare un riavvio ritardato ignorando i passaggi eseguiti in precedenza. Ciò fornisce un maggiore controllo sull'elaborazione in batch utilizzando checkpoint persistenti a livello di fase.
- AS400: Supporto aggiuntivo alla libreria, prestazioni e robustezza migliorate dei comandi di uso comune come CPYF, OVRDBF, SBMJOB e OPNQRYF e molti altri.

Abbiamo testato questa versione di Transform for mainframe Runtime con lo stack seguente. AWS Potrebbero essere compatibili anche altre versioni dei componenti.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 18.18
	Npm 9.8
	Angolare 17
Livello di servizio	Spring Boot 3.2.5
	Spring Core 6.1.5
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

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

Versione di runtime 4.3.0

ZoS

Nuove funzionalità

- CICS - Aggiunto il supporto per il riferimento alla mappa nel comando SEND MAP
- CICS - Aggiunto il supporto per il comando RECEIVE e il supporto per l'esecuzione di transazioni con dati dallo schermo `JicsTransactionRunner`
- È stato aggiunto il supporto per l'intestazione IIH per i messaggi JMS
- COBOL - Aggiunto il supporto per più spazi incorporati nello pseudo-testo per l'istruzione REPLACING
- COBOL - Aggiunto il supporto per l'istruzione JSON PARSE
- <noloc>Blusam</noloc> - Aggiunto il supporto per KMS alla funzionalità «Esporta set di dati»
- BAC - Aggiunta la configurazione `application-main.yaml` per definire la dimensione del record per filtrare le maschere caricate corrispondenti a questa dimensione del record
- JCL - INFUTILB - Aggiunto il supporto per la parola chiave INTO come parte dell'istruzione di controllo BMC
- GS21 - Aggiunta la gestione SOSI per la codifica JEF
- GS21 - JCL - Aggiunto KDJBR14 come alias di IEFBR14
- GS21 - JCL - Aggiunto KQCAMS come alias di IDCAMS
- MF - Aggiunto il supporto per i file compatibili con COBOL MF in base al supporto sul campo
- MF - Aggiunto il supporto per il meccanismo SORT per i file compatibili con COBOL MF
- MF - Aggiunto il supporto per il file mancante aperto non opzionale compatibile con COBOL MF

Miglioramenti

- JCL - DSNUTILB - Funzionamento di caricamento migliorato con ZONED DECIMAL Type
- JCL - DSNUTILB - Aggiunto il supporto per la chiave duplicata
- JCL - DSNUTILB - Aggiunto il supporto per il meccanismo di rollback sul comando LOAD
- JCL - INFUTILB - UNLOAD migliorato con nuove proprietà FETCHSIZE e CHUNKSIZE

- JCL - A - IKJEFT1 Lettura migliorata dei file SYSTSIN aggiungendo il set di caratteri corrente
- JCL - DFSORT - Aggiunto il supporto per l'opzione & DATE4 DATE5
- JCL - DFSORT - Aggiunto il supporto per i casi di tipo Variable Block come input e Fixed Block come output
- JCL - DFSORT - Aggiunto il supporto per ALTSEQ
- JCL - Metadati di checkpoint migliorati con identificatore web del lavoro
- JCL - Eliminazione migliorata del checkpoint di riavvio del Batch per REDIS
- IMS - Funzione EXPRESS implementata per il comando PURGE
- IMS - Aggiunto il supporto per le opzioni PCBNAME e LIST for PCB statement
- COBOL - Aggiunto il supporto per l'istruzione GO TO senza target
- CICS - Supporto migliorato per l'istruzione INTO con READQ RecordAdaptable TS
- CICS - Supporto migliorato per il comando INQUIRE TRANSACTION
- CICS - Supporto migliorato per SetBytes nel comando READNEXT
- CICS - Supporto migliorato per il comando START senza l'opzione CHANNEL
- CICS - Aggiunto il supporto per il tipo di riferimento per Inquire TSQueue
- CICS - Supporto migliorato per il comando RECEIVE MAP quando map e mapset sono Reference
- CICS - Supporto migliorato per le opzioni FROM e LENGTH per il comando RECEIVE MAP
- CICS - Aggiunto il supporto dell'attributo RecordAdaptable
- CICS - Supporto migliorato per il comando RECEIVE per gestire l'overflow
- CICS - Aggiunto il supporto per la regola slice nelle istruzioni CICS
- CICS - Supporto migliorato per le strutture di collegamento DFHCOMMAREA e DFHEIBLK. Il motore di trasformazione supporta definizioni più implicite
- CICS - Aggiunto il supporto per le opzioni START, NEXT e END per il comando INQUIRE CONNECTION
- CICS - Aggiunto il supporto sia per il tipo 'int' che per 'reference' per l'opzione LENGTH del comando RECEIVE
- CICS - Supporto migliorato per l'analisi del comando INQUIRE NETNAME
- CICS - Aggiunto il supporto per il nome del gruppo per JicsQueueBuilder
- <noloc>Blusam</noloc> - Aggiunto il supporto per i file indicizzati che iniziano con una chiave generica

- <noloc><noloc>Blusam - Caricatori Blusam migliorati</noloc></noloc>
- BAC - Supporto migliorato per la sincronizzazione dei dati in ambienti multiistanza quando Redis viene utilizzato per centralizzare i valori memorizzati nella cache, inclusi i dati effettivi e i blocchi
- BAC - Interfaccia utente migliorata (stile, logo, casella di controllo)
- BAC e JAC: aggiunta la configurazione di `application-main.yaml` per recuperare il nome utente e la password dell'utente super amministratore predefinito nel segreto di AWS Secrets Manager specificando l'ARN
- BAC e JAC - Aggiornamento della dipendenza a Bootstrap 5
- Checkpoint JCL migliorati e configurazione del modello JICS Redis TSQueues
- Supporto migliorato per Size of Pointer a seconda di AMode
- È stato aggiunto il supporto per il confronto zero su NumericEditedType
- Proprietà MDC SLF4j applicate prima della registrazione
- Supporto migliorato alla lettura dei file su più righe vuote gestite
- MF - Supporto migliorato per l'inizializzazione delle variabili di puntamento per la direttiva del compilatore COBOL MF InitPtr
- Redis - Funzionalità migliorata sull'aspetto della concorrenza attraverso un'implementazione basata su Redisson GwFileLock

AS400

Nuove funzionalità

- CL - Aggiunto il supporto per il comando CHGPF
- RPG - Aggiunto il supporto per le funzioni %HOURS, %MINUTES e %SECONDS
- COBOL - Aggiunto il supporto del file SORT con l'architettura DAO Blu4iv

Miglioramenti

- CL - Migliorato PgmClose per essere registrato come programma e accettare una varietà di oggetti per il parametro OPNID
- CL - RTVMBRD rifattorizzato per gestire più librerie e membri
- CL - Aggiunto il supporto per il parametro TOLIB sul comando MOVOBJ
- CL - Supporto migliorato della partizione sul comando CPYFRMSTMF

- CL - Aggiunto il supporto per il parametro SNDMSG TOUSR
- CL - Supporto migliorato del comando OVRDBF
- CL - Prestazioni migliorate per il comando OVRDBF - Aggiornamento dei valori predefiniti per srcfile e member
- CL - Copia dei file migliorata con il comando CPYF
- CL - Comando CPYF riprogettato per essere più robusto e gestire meglio QTEMP, CRTFILE, FROMRCD & TORCD, MBROPT e FMTOPT (MAP & DROP)
- CL - Supporto migliorato per il comando CPYF nei casi in cui FROMFILE e TOFILE hanno colonne non corrispondenti
- CL - Migliore la gestione da parte di CPYF NOCHK delle colonne con nomi diversi quando viene specificato REPLACE
- CL - Aggiunta un'implementazione vuota per il comando CRTDUPOBJ sui file logici
- CL - Gestito il problema di indicizzazione delle sottostringhe con il comando CHGDTAARA
- CL - Supporto migliorato del comando SBMJOB
- CL - Made OverrideManager e OpnqryfHelper mapping non fanno distinzione tra maiuscole e minuscole
- Schermo: è stata migliorata la messa a fuoco iniziale del primo campo modificabile quando non è specificato un cursore
- Schermo: posizione di messa a fuoco migliorata dopo la chiusura e durante l'utilizzo del menu di aiuto
- Schermata: è stata migliorata la messa a fuoco del cursore dopo aver premuto la pagina up/down nel componente della tabella
- Schermo: supporto migliorato per messaggi di errore e messa a fuoco su più campi
- Schermata: calcolo migliorato del numero di riga per i campi dei sottofile
- Schermata: supporto migliorato per i file secondari inizializzati utilizzando SFLINZ
- Screen - Supporto migliorato per l'immissione solo numerica
- Schermata - Migliore gestione della parola chiave WINDOW in DSPF con 3 parametri
- Schermata: posizione migliorata del piè di pagina per la tabella con record contenenti più di 1 riga
- Schermata - Navigazione della pagina migliorata per il messaggio di rotazione che rimane sulla pagina su/giù
- Funzionalità EDITC migliorata per modificare il codice 3

- Meccanismo di blocco dell'area dati Blu4iv migliorato per non fare nulla quando non ci sono blocchi da sbloccare invece di generare eccezioni
- È stato aggiunto il supporto per restituire il numero di righe interessate in StraightQueryBuilder
- Meccanismo di log QTEMP migliorato
- Migliorato DAOManager reads/writes/deletes per i casi d'uso su file sovrascritti da una libreria di file + diversa

Funzionalità trasversali

Nuove funzionalità

- È stato aggiunto un modo centralizzato per gestire le proprietà di sistema relative a SSL/TLS tramite configurazione, che consente l'uso di Gestione dei segreti AWS
- Configurazione avanzata delle risorse IBMMQ con Gestione dei segreti AWS
- JCL - Aggiunta la configurazione della posizione temporanea per Runtime, ha risolto i file groovy tramite la proprietà YML tempFilesDirectory e ha aggiunto la possibilità di specificare se eliminare il contenuto della cartella dei file temporanei all'avvio dell'applicazione tramite la proprietà YML cleanTempFiles DirectoryAtStartup
- Aggiungi segreti AWS per tutte le credenziali Redis

Miglioramenti

- Conversione migliorata dal tipo alfanumerico al tipo numerico modificato
- Controllo migliorato DataUtils: :isNumeric per PackedType
- Timestamping migliorato dei file di registro
- Accesso separato gestito. ZonedType decodeAsString
- COBOL - Supporto migliorato della dichiarazione INITIALIZE
- Supporto migliorato per. DataUtils compareAlphInt per gestire gli spazi iniziali e finali per AS400 ZOS
- SQL - Migliore convalida implicita del runtime del cursore in sola lettura
- SQL: meccanismo di memorizzazione nella cache dei metadati migliorato
- Rimuovi la connessione al database <noloc>Jics/Blusam</noloc> dall'applicazione Gapwalk `application-main.yml`

Versione 4.3.0 degli strumenti di modernizzazione

ZoS

Nuove funzionalità

- GS21 - Aggiunto il supporto per COBOL CONSTANT SECTION GS21
- GS21 - Aggiunta la codifica JEF ai set di caratteri disponibili

Miglioramenti

- CICS - Aggiunto il supporto per l'analisi del comando DOCUMENT CREATE
- CICS - Aggiunto il supporto per l'analisi del comando CICS WEB EXTRACT
- CICS - Aggiunto il supporto per l'analisi del comando WEB WRITE
- CICS - Aggiunto il supporto alla trasformazione per DB2 CONN SIGNIN e PLAN
- CICS - Supporto migliorato per l'analisi del comando SEND MAP ignorando l'opzione TERMINAL
- CICS - Supporto migliorato per l'analisi del comando RETURN ignorando l'opzione ENDACTIVITY
- MFS - Supporto migliorato per generare file MFS con estensione specifica
- COBOL - Supporto migliorato per l'istruzione REPLACE
- COBOL - Percorso dinamico gestito e direttiva del compilatore MF
- COBOL - Supporto migliorato per il valore OMESSO nella dichiarazione CALL
- COBOL - Accesso migliorato ai campi multidimensionali per supportare il valore firmato
- COBOL - Aggiunto il supporto per la clausola OF per l'istruzione FILE STATUS
- COBOL - Migliore analisi della dichiarazione RESULT-SET-LOCATOR
- JCL - IDCAMS - Aggiunto il supporto per l'abbreviazione RECORDS

AS400

Nuove funzionalità

- CL - Aggiunto il supporto per variabili definite e basate su puntatori nella trasformazione CL
- CL - Aggiunto il supporto per caratteri speciali in DCLF
- È stato aggiunto il supporto per l'API retrieve call stack (QWVRCSTK)

Miglioramenti

- RPG - Migliore trasformazione dei parametri della procedura utilizzando la parola chiave `likeds`
- RPG - Supporto per la revisione della parola chiave EXTNAME
- RPG - Valore letterale di supporto migliorato*ALL
- RPG - Supporto migliorato per le specifiche di output e i file descritti dal programma
- DDS - Risoluzione migliorata dei campi DDS in un LF che fa riferimento a un PF che fa riferimento a un dizionario PF
- Schermata: indicatori cancellati quando si utilizza l'istruzione CLEAR per cancellare un record da DSPF
- CL - Miglioramento transformation/generation dei parametri CL con elenchi di elementi

Capacità trasversali

Miglioramenti

- SQL - Migliore generazione di query SQL contenenti N con carattere tilde
- COBOL - Supporto migliorato dell'istruzione LENGTH OF per i campi di gruppo
- COBOL - Supporto migliorato per i campi REDEFINED utilizzando quaderni

Note di rilascio 4.2.0

Data di rilascio: 10 luglio 2024

Questa versione di AWS Transform for mainframe Runtime and Modernization Tools è incentrata su prestazioni e sicurezza. Alcune funzionalità e modifiche chiave di questa versione sono:

- Abbiamo migliorato le prestazioni di trasformazione, in particolare per progetti di grandi dimensioni con oltre 30 milioni di righe di codice. Abbiamo implementato una serie di miglioramenti e i risultati ottenuti hanno mostrato una riduzione dei tempi di oltre il 150% e le esecuzioni sono state completate in pochi minuti anziché ore. Il miglioramento chiave che abbiamo implementato è la configurazione di un meccanismo di timeout per limitare il tempo massimo assegnato all'analisi in modo da saltare i file con problemi rilevati. Contrassegniamo i file ignorati in modo da poterli esaminare in un secondo momento, se necessario.
- Abbiamo aggiunto il supporto per un sistema distribuito di gestione dei blocchi per i AS400 progetti. In un ambiente ad alta disponibilità (multinodo) in cui più istanze dell'applicazione sono destinate

allo stesso database, mantenere la coerenza dei dati durante l'intero ciclo di vita di queste istanze è una sfida significativa. Per affrontare efficacemente questa sfida, abbiamo aggiunto Redis come server di caching condiviso ed esterno per coordinare tutte le istanze durante l'esecuzione in modalità batch.

- Abbiamo aggiunto una nuova funzionalità di impaginazione dinamica per il componente tabella. L'obiettivo di questa funzionalità è migliorare i tempi di risposta e ridurre l'utilizzo della memoria per le tabelle con un numero elevato di righe. Questa funzionalità consente al componente tabella di caricare solo una parte dei dati e di recuperare più record su richiesta mentre navighi tra le pagine. Per migliorare ulteriormente l'esperienza, la piattaforma supporta anche il prerecupero dei dati. Questa nuova funzionalità di impaginazione dinamica offre un'esperienza utente più efficiente e reattiva per le applicazioni con set di dati di grandi dimensioni.
- Per affrontare una sfida fondamentale che si presenta spesso, abbiamo aggiunto il supporto per i programmi COBOL annidati. In precedenza, la soluzione alternativa per modernizzare i programmi COBOL annidati prevedeva la separazione manuale dei programmi in file diversi, il collegamento tramite la sezione di collegamento e la loro chiamata reciproca con gli argomenti necessari. Questo processo non solo richiedeva molto tempo, ma era anche soggetto a errori. Ora è possibile modernizzare i programmi COBOL annidati senza la necessità di separarli manualmente.

Abbiamo testato questa versione di AWS Transform for mainframe Runtime con lo stack seguente. Potrebbero essere compatibili anche altre versioni dei componenti.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 18.18
	Npm 9.8
	Angolare 17
Livello di servizio	Spring Boot 3.2.4
	Spring Core 6.1.5
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14

	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

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

Versione di runtime 4.2.0

ZoS

Nuove funzionalità

- DB2 - È stato aggiunto il supporto per l'invocazione di procedure memorizzate senza qualificatore di schema nella query SQL
- COBOL - Aggiunto il supporto per la funzione HEX-OF
- COBOL - Aggiunto il supporto per i programmi annidati
- COBOL - Aggiunto il supporto per FUNCTION e TEST-DATE-YYYYMMDD TEST-DAY-YYYYDDD
- CICS - Aggiunto il supporto per l'opzione UCTRANST nel comando SET TERMINAL
- CICS - Aggiunto il supporto per il comando INQUIRE CONN DB2
- <noloc>Blusam</noloc> - Aggiunto il supporto per l'eliminazione delle chiavi su VSAM ad accesso dinamico
- IMS - Aggiunto il supporto per il comando TERM
- BAC - Aggiunti controlli di autorizzazione su tutti gli endpoint BAC REST
- BAC - È stata aggiunta la configurazione `application-main.yaml` per definire una dimensione del record per filtrare le maschere caricate che corrispondono alla dimensione del record
- BAC e JAC: è stata aggiunta la configurazione tramite la quale è possibile `application-main.yaml` recuperare il nome utente e la password dell'utente super amministratore predefinito nella forma segreta `command` specificando l'ARN

Miglioramenti

- JCL - SORT - Supporto migliorato per la clausola OMIT per gestire le condizioni con Shiftin e caratteri ShiftOut
- JCL - SORT - Supporto migliorato per il campo BDW
- JCL - SORT - Supporto migliorato per più concatenazioni GDG con il campo BDW

- JCL - DFSORT - Aggiunto il supporto per le clausole INREC PARSE STARTAFT /STARTAT
- JCL - IEBGENER - Gestione RecordSize migliorata per i file di output
- JCL - INFUTILB - INDICATORE NULL disabilitato basato su YML-FIX GRAPHIC CASE
- JCL - Supporto migliorato per la gestione delle costanti nel campo FormatterParser OUTREC
- JCL - Dati di caricamento migliorati per il tipo di grafica nell'utilità del programma DSNUTILB
- JCL - SORT - Supporto migliorato per il formato decimale suddiviso in zone
- JCL - SORT - Supporto migliorato per la clausola OMIT per gestire le condizioni con Shiftin e caratteri ShiftOut
- MQ - Migliorata la gestione della connessione MQ per adattarsi a diversi flussi di lavoro aziendali
- CICS - Supporto migliorato del riferimento ai puntatori per le istruzioni EXEC CICS READ SET (ptr-ref)
- COBOL - Supporto migliorato per il record della sezione di collegamento ADDRESS OF
- COBOL - Aggiunto il supporto per EXP e funzioni EXP10
- COBOL - Supporto migliorato per l'istruzione REPLACE tramite copybook
- COBOL - Accesso ai campi multidimensionale migliorato per supportare i valori firmati
- MF COBOL - Aggiunto il supporto per file sequenziali in formato variabile
- IMS - Lettura migliorata della configurazione dei file YML IMS per rendere possibile l'utilizzo di variabili di ambiente
- IMS - Sono stati gestiti altri modi per specificare il numero del segmento
- IMS - Maggiore robustezza quando un programma IMS viene chiamato da una transazione avviata programmaticamente
- IMS - Migliorati i criteri di ricerca SSA build per tenere conto della lunghezza corrente della clausola WHERE se non viene fornita la lunghezza implicita del segmento
- IMS - Lettura migliorata della configurazione dei file IMS YML per consentire l'uso di variabili di ambiente
- Supporto migliorato per la clausola VALUE in NumericEditedType
- Supporto migliorato per la concatenazione di stringhe per gestire il caso in cui la prima stringa da concatenare sia vuota, vuota o con spazi

AS400

Nuove funzionalità

- È stato aggiunto il supporto per l'impaginazione all'interno del componente Table; i progetti possono utilizzare questa funzionalità per ridurre il tempo e le dimensioni di risposta quando viene caricato un componente Table con un numero elevato di righe
- È stato aggiunto il supporto delle librerie per le query SQL sull' AS400 applicazione; poiché le librerie vengono convertite in partizioni nelle applicazioni moderne, abbiamo adattato il runtime per riscrivere le query di conseguenza
- RPG - Aggiunto il supporto per la libreria QTEMP per le query SQL
- RPG - Aggiunta la codifica nella funzione CONVERT per gestire valori di input vuoti
- RPG - Aggiunto il supporto per le funzioni %HOURS, %MINUTES e %SECONDS
- CL - Aggiunto il comando CHGPFM
- CL - È stato aggiunto il supporto per la parola chiave *FROMLIB nel comando CRTDUPOBJ
- CL - Aggiunto il supporto per la creazione di tabelle e partizioni per nomi di tabelle che superano i 9 caratteri
- CL - Aggiunto il supporto per l'eliminazione di file flat nelle sottocartelle per il comando DLTF

Miglioramenti

- Schermata: è stato migliorato ErrorMessage il collegamento a un campo specifico e l'aggiunta a ArrayMessageLine
- Schermata: cursore errormsg migliorato
- Schermo: migliorato ArrayMessageLine per non essere incluso nell'ordine delle schede
- Schermo: visualizzazione migliorata degli array di messaggi di errore sullo schermo AS400
- SQL - Supporto del cursore migliorato per eseguire il commit della transazione alla chiusura per evitare blocchi nella creazione delle partizioni
- CL - Aggiunto il supporto per il PgmCall comando e migliorato il pattern non supportato da QCMDEXC
- CL - Supporto migliorato per il comando CHKOBJ per gestire OBJTYPE PGM
- CL - Supporto multilibreria migliorato per CPYF e altri comandi CL che gestiscono librerie e partizioni
- CL - Aggiunto il supporto per il passaggio di una variabile relativa al nome del programma nel comando CALL PGM
- CL - Ha risolto il caso del tipo predefinito di tipo di oggetto
- CL - Aggiunto il supporto multilibreria per il comando CRTDUPOBJ

- CL - Gestione migliorata delle connessioni al database su più comandi
- CL - Supporto migliorato per RMVLNK per gestire il caso in cui un file o una directory non vengono trovati e il messaggio di monitoraggio CPF0000
- CL - CLRPFM migliorato per tenere conto della libreria durante la rimozione dei record
- CL - CPYF - Comando migliorato per supportare la libreria QTEMP, il parametro FmtOpt (*NoChk) e il carattere di controllo
- CL - Gestione fissa delle virgolette e dei parametri mancanti nei comandi RMVLNK e CPY
- RPG - Scoping variabile migliorato; ora rientra nell'ambito di lavoro anziché nell'ambito di DataArea collegamento
- RPG - Interrogazioni di lettura DAO migliorate da eseguire senza transazione per evitare situazioni di stallo
- Ricerca migliorata della messaggistica MQ aggiungendo un trim a MSGQ sulla ricerca DB
- Sono state rimosse le dichiarazioni di transazione non necessarie sul supporto della connessione al database
- Migliorato l'aggiornamento dello stato del lavoro di Quartz in caso di eccezione
- Aggiunto il supporto per gestire il caso in cui un array di indicatori non è inizializzato

Funzionalità trasversali

Nuove funzionalità

- Redis: aggiunta la configurazione Redis globale per tutte le cache Redis
- Aggiunta funzionalità di tracciamento della sessione per consentire di archiviare le informazioni di tracciamento della sessione (ID di sessione, nome utente associato, timestamp di creazione e ID del nodo) rendendo persistenti i dati su Redis
- È stata aggiunta la configurazione temporanea della posizione per i file groovy risolti in fase di esecuzione tramite la proprietà YML`tempFilesDirectory`; è stata inoltre aggiunta la possibilità di specificare se eliminare il contenuto della cartella dei file temporanei all'avvio dell'applicazione tramite la proprietà YML `cleanTempFilesDirectoryAtStartup`

Miglioramenti

- Supporto migliorato per le proprietà di configurazione dell'implementazione del pool di connessioni per le fonti di dati di utilità

- Supporto migliorato per la modalità stampante e il controllo del carrello ANSI basato sull'utilizzo delle clausole ADVANCING e delle clausole WRITE BEFORE
- Versione Angular aggiornata sull'applicazione front-end per progetti modernizzati
- Costruzione avanzata della sintassi degli URL del gestore segreto per DB2
- È stato migliorato il. DataUtils compareAlphInt metodo per aggiungere il supporto per gli spazi finali
- Supporto SQL migliorato per l'output di tipo blob
- Maggiore robustezza per i job trigger tramite endpoint post/script

Versione 4.2.0 degli strumenti di modernizzazione

ZoS

Nuove funzionalità

- CICS - Aggiunto il supporto per l'analisi dei comandi WEB CICS
- CICS - Aggiunto il supporto per la trasformazione del comando MONITOR
- CICS - Aggiunto il supporto per l'analisi del comando CICS SEND MRO
- COBOL - Aggiunto il supporto per l'analisi dell'istruzione NO REWIND
- COBOL - Aggiunto il supporto per il tipo di numero dell'opzione UCTRANST nel comando CICS SET TERMINAL
- COBOL - Aggiunto il supporto per la clausola MULTIPLE FILE in I-O-SECTION
- CSD - Aggiunto il supporto per la trasformazione di più file CSD
- CSD - Aggiunto il supporto per la generazione di jicsFileAix .json da più file CSD
- IDCAMS - Aggiunto il supporto per la creazione di un relativo set di dati di record (RRDS)

Miglioramenti

- Prestazioni migliorate durante il calcolo delle maschere SQL
- COBOL - Analisi migliorata dell'inutile clausola RESERVE in FILE-CONTROL
- COBOL - Migliore analisi di SECTION e CLASS
- COBOL - Gestione DFHRESP migliorata
- COBOL - Supporto migliorato per EXIT PARAGRAPH tramite perform
- IMS - Supporto migliorato per i nomi dei segmenti specificati utilizzando doppie parentesi

- IMS - Ha arricchito la generazione di codici di stato quando vengono richiamati SCHD e TERM
- COBOL - Migliore generazione di campi DEPENDING ON
- COBOL - Trasformazione migliorata della funzione integrata DB2 TO_TIMESTAMP

AS400

Nuove funzionalità

- Aggiunto il supporto per la conversione di campi alfanumerici come CHAR negli script SQL
- COBOL400 - È stato aggiunto il supporto per i file DATABASE descritti dal programma

Miglioramenti

- DDS - Supporto migliorato per il nome ALIAS
- Supporto migliorato per il tipo float senza valore iniziale
- COBOL 400 - Calcolo delle dimensioni migliorato per il tipo suddiviso in zone con segno

Funzionalità trasversali

Miglioramenti

- Migliore segnalazione degli ID di errore relativi all'analisi DDS e SQL
- Generazione di codice migliorata sui rami delle condizioni
- Prestazioni migliorate nella generazione di bollettini meteorologici

Note di rilascio 4.1.0

Data di rilascio: 31 maggio 2024

Questa versione di AWS Transform for mainframe Runtime and Modernization Tools è incentrata su prestazioni e sicurezza. Alcune funzionalità e modifiche chiave di questa versione sono:

- Trasformazione e prestazioni: per consentire ai progetti con un'ampia base di codice (+50 milioni di righe di codice) di trasformarsi con successo, abbiamo ottimizzato le prestazioni e l'ingombro di memoria dell'intero meccanismo di trasformazione.
- BAC/JAC: la sicurezza è la massima priorità. AWS Le applicazioni modernizzate con AWS Transform for mainframe devono essere conformi agli standard di sicurezza. Abbiamo apportato

alcuni importanti aggiornamenti alla <noloc>Blusam</noloc> Administration Console (BAC) e alla JICS Administration Console (JAC) per renderle più sicure:

- È stata aggiornata l'applicazione ad Angular v17.
- Oltre al supporto nativo per AWS Cognito, abbiamo aggiunto il supporto generico OAuth che consentirà una maggiore flessibilità nel consentire ai clienti di utilizzare il provider di identità di loro scelta.
- Abbiamo configurato ed esteso le funzionalità di sicurezza utilizzando intestazioni appropriate.

- AS400 - Supporto multinodo per il meccanismo di blocco del database. Ha fornito la possibilità di collegare un server di caching condiviso ed esterno (Redis) per eseguire un'applicazione batch su più istanze, come la modernizzazione gestita del mainframe. AWS

Questa versione del AWS Transform for mainframe runtime è stata testata con lo stack seguente. Potrebbero essere compatibili anche altre versioni.

Componente	Versione testata
Java	Java 17
Livello di presentazione	Nodo JS 18.18
	Npm 9.8
	Angolare 16.1
Livello di servizio	Spring Boot 3.2.5
	Spring Core 6.1.5
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21c
Server applicazioni	Apache Tomcat 10.1.17

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

Versione di runtime 4.1.0

ZoS

Nuove funzionalità

- Configurazione aggiunta per la gestione dinamica dei OAuth2 provider. Introdotto SECRET_ OAUTH2 _PROVIDER_NAME_KEY per specificare il provider. Metodo di recupero segreto aggiornato per gestire più provider. I segreti garantiti vengono recuperati in modo sicuro da. Gestione dei segreti AWS
- È stato aggiunto il supporto per le proprietà DB2 SSL per consentire di definire un certificato SSL (sslTrustStoreposizione) e una password (sslTrustStorepassword) per sbloccare il file keystore. Gestione dei segreti AWS
- È stato aggiunto il supporto per fonti di dati aziendali esterne.
- JCL - Aggiunto il supporto per il meccanismo di checkpoint per il riavvio in batch.
- JCL - Aggiunto il supporto per i parametri DCB, la dimensione del record e l'RDW.
- JCL - Aggiunta la configurazione dinamica del nome della cartella per i file temporanei generati.
- REDIS - Aggiunta la configurazione del pool nella configurazione Redis per JICS.
- REDIS - Aggiunto l'indice del database nella configurazione Redis per Catalog e JICS.
- BatchScript - Aggiunta la propagazione del nome del passaggio per l'esecuzione delle esecuzioni dei programmi.
- CICS - Aggiunto il supporto per il comando ADDRESS SET.
- CICS - Aggiunto il supporto per PURGE MESSAGE e JUSTIFY.

Miglioramenti

- JCL - INFUTILB - Supporto migliorato per la disabilitazione dell'indicatore null basato sulla proprietà YML.
- JCL - INFUTILB - Supporto migliorato per il tipo di dati. CHAR/BPCHAR
- JCL - ICEGENER - Aggiunto il supporto per la copia di flussi di input multilinea nei file.
- JCL - IEBGENER - Supporto migliorato per la gestione della conversione da file a blocchi variabili a blocchi fissi.
- JCL - DFSORT - Supporto migliorato per i parametri a più cifre sull'operazione DATE.
- JCL - DFSORT - Aggiunto il supporto per la clausola INCLUDE=ALL.

- JCL - Supporto migliorato per l'utilità SORT per gestire il campo BDW in output.
- JCL - Supporto migliorato per la concatenazione DD.
- JCL - Supporto migliorato per Input Stream.
- JCL - DSNUTILB - Supporto migliorato per l'istruzione NULLIF ().
- JCL - INFUTILB - Aggiunto il supporto per lo scaricamento dei dati con l'opzione NOPAD.
- JCL - INFUTILB - Supporto migliorato per la data corrente in INFUTILB.
- JCL - Aggiunti i controlli dell'esistenza e delle dimensioni dei file prima di utilizzare un file.
- JCL - GDG - Migliorata la gestione delle sottodirectory per GDG.
- MQ - Apertura della connessione migliorata nell'implementazione JMS.
- MQ - Migliore impostazione della lunghezza dei dati del messaggio GET per l'origine dati XA.
- MQ - Quaderno standard CMQV scomposto per prevenire errori di compilazione e utilizzi di refactoring.
- <noloc>Blusam</noloc> - Supporto migliorato per le richieste di eliminazione di set di dati inesistenti.
- Supporto migliorato per l'istruzione ALLOCATE.
- Migliore robustezza della denominazione TS-QUEUE.
- BatchScript - Conservazione migliorata del codice di ritorno del passaggio precedente nella riesecuzione del lavoro.
- Set di dati: è stato migliorato il controllo dell'esistenza dei file quando un file esiste ed è temporaneo.
- Set di dati: è stata migliorata la concorrenza nella localizzazione dei file GDG da eliminare.
- Set di dati: aggiunto il supporto per ottenere le dimensioni record del set di dati GDG.
- CICS - Supporto migliorato per l'opzione SUSPENDED nel comando INQUIRE TASK LIST.
- CICS - Supporto migliorato per LOAD SET utilizzando l'istruzione ADDRESS OF.
- CICS - Argomenti CICS non gestiti REMOTESYSTEM migliorati quando CICS INQUIRE.
- CICS - Supporto migliorato per il comando GETMAIN per gestire l'opzione SET con un puntatore definito con la parola chiave OF.
- JICS - Migliore robustezza del metodo jics XAPrepare () aggiungendo il controllo dello stato della transazione.
- JICS XA - Aggiunto un controllo dello stato della transazione e una terminazione migliorata del thread di transazione.

- BAC - Autenticazione avanzata basata sui ruoli sul lato client e su tutte le chiamate API. refactored/centralized
- BAC - Implementata una funzionalità per bloccare l'accesso pubblico a BAC e JAC in base alla configurazione
- BAC - Aggiornamento delle dipendenze: Angular 17.
- BAC - Migliore integrazione della sicurezza con. OAuth2
- BAC - DDL migliorata generata dall'ibernazione.
- BAC - Meccanismo migliorato del set di dati di esportazione.
- JAC - Aggiornato ad Angular 17 e riporta tutte le specifiche del lavoro da BAC (ROLE, sadmin conf, XSRF, logout).
- COBOL - Aggiunto il supporto per le funzioni CHAR e ORD-MIN.
- Migliorato FileFactory per mantenere le dimensioni dei record del catalogo nella disposizione MOD.
- Registrazione abilitata tramite MDC per transazioni JICS.
- SQLCA > SQLSTATE migliorato prodotto per le stored procedure che generano set di risultati ad hoc.
- Supporto migliorato per la pianificazione delle attività relative all'ultimo aggiornamento di Spring.

AS400

Nuove funzionalità

- È stato aggiunto il supporto multinodo per i blocchi dei record del database utilizzando Redis.
- È stato aggiunto il supporto per BINARY CHARACTER per il tipo DDS.
- CL - Aggiunto il supporto per la generazione di file di report personalizzati.
- RPG - Aggiunto il supporto per la parola chiave RENAME sui primary/secondary file.

Miglioramenti

- Supporto del database migliorato per la gestione della colonna CTID con una clausola JOIN.
- Posizione del cursore migliorata per più DSPATR (PC).
- Registrazione migliorata delle eccezioni di lettura.
- Registrazione dei lavori Quartz migliorata per includere le proprietà dei lavori in MDC.

- Supporto migliorato per la schermata di aiuto. AS400
- CL - Supporto migliorato per il comando RMVJOBSCDE per accettare numeri di immissione con spazi finali.
- CL - Supporto migliorato per il comando RMVJOBSCDE per rimuovere una pianificazione dei processi utilizzando un nome di lavoro generico.
- CL - Supporto migliorato per il comando SAVOBJ per ordinare i record per chiave di tabella.
- CL - Supporto migliorato per il comando CPYF per stabilire una nuova connessione per le query DB.
- CL - Inserimento migliorato dei messaggi di richiesta nei messaggi di coda con SNDPGMMSG.
- CL - Configurazione della coda di lavoro migliorata per specificare la coda di lavoro predefinita.
- CL - Migliorato il comando CRTPF per supportare la libreria QTEMP e il parametro RCDLEN.
- CL - Supporto migliorato per il comando CHKOBJ - Verifica la presenza di partizioni con libreria.
- CL - RTVMGS migliorato da inviare e quando non viene trovato. CPF2407 CPF2419 file/ID
- CL - Interpretazione migliorata da CPYTOIMPF e CPYFRMIMPF dei parametri di formattazione precedenti.
- CL - Aggiunto il supporto per il parametro OVRPRTF USRDTA.
- CL - Migliorato il comando CL CPYTOIMPF per stabilire una nuova connessione ed evitare di chiudere i set di risultati esistenti.
- CL - CHGDTAARA migliorato in modo che non modifichi più la lunghezza dell'area dati quando aggiorna il contenuto.
- CL - Migliore ClCommand gestione delle connessioni al database.
- Interazione ottimizzata tra front-end e backend.
- COBOL - Trasformazione aggiornata per gestire FILLER nei quaderni.
- Visualizzazione migliorata delle informazioni aggiuntive sui messaggi per i messaggi personalizzati inviati al front-end.
- Aggiornato il valore predefinito per il selettore in app.component.ts.
- È stata migliorata la suddivisione del testo nella visualizzazione. split-dynamic-field
- È stata migliorata la visualizzazione del messaggio di errore con più scritture seguite da una lettura.

Funzionalità trasversali

Nuove funzionalità

Aggiunto il supporto per la configurazione dinamica del OAuth2 provider secret.

Miglioramenti

- Stampa: supporto migliorato dei parametri QCMDEXC per la gestione delle virgolette e migliore formazione dei nomi dei report
- Supporto migliorato per la sintassi delimitata su. RecordAdaptable
- Registrazione InspectBuilder degli errori migliorata per aggiungere contesto sulla stringa di origine.
- DataSimplifier - maggiore robustezza per l'affettazione. ByteArray
- Registrazione MDC migliorata con nuovi attributi di runtime.

Versione 4.1.0 degli strumenti di modernizzazione

ZoS

Nuove funzionalità

- È stato aggiunto il supporto per più trasformazioni di file CSD
- COBOL - Aggiunto il supporto per l'istruzione CICS ALLOCATE.
- COBOL - Aggiunto il supporto per ON SIZE ERROR nell'istruzione ADD CORRISPETTING.
- COBOL - Aggiunto il supporto per EXIT PARAGRAPH.

Miglioramenti

- COBOL - Supporto migliorato per il copybook -INC.
- COBOL - Supporto migliorato per l'inizializzazione di FILLER.
- COBOL - Supporto migliorato per il confronto dei valori figurativi.
- COBOL - Supporto migliorato per WHEN ANY nelle clausole WHEN consecutive prive di blocchi di codice intermedi.
- COBOL - Supporto migliorato per la costante figurativa.
- COBOL - Supporto migliorato per il calcolo delle dimensioni dei tipi compressi.
- COBOL - Argomento CICS migliorato e non gestito KEEP per SPOOLCLOSE.
- COBOL - Generazione migliorata per la funzione TEST-NUMVAL.
- COBOL - Argomenti di generazione Java migliorati sul supporto del framework INSPECT.

- CICS - Supporto migliorato per la definizione di DFHCOMMAREA.

AS400

Nuove funzionalità

- RPG - Aggiunto un meccanismo di rilevamento degli errori per generare il DDS (incompleto) in modo da non bloccare la generazione del programma.
- È stato aggiunto il supporto per la parola chiave di specificazione della descrizione del file INCLUDE.

Miglioramenti

- RPG - Analisi migliorata e completamente gratuita.
- RPG - Maggiore robustezza con rilevamento degli errori.
- RPG - Inizializzazione migliorata della parola chiave with export. field/DS
- RPG - Funzionamento DAO migliorato per gestire gli indicatori.
- RPG - Gestito il valore predefinito di PERRCD con CTDATA.
- RPG - È stato aggiornato il parser Free-RPG per registrare un errore unico per ogni regola di analisi.
- PRTF - Ha gestito la collisione dei nomi tra PRTF e JRXML.
- COBOL - Supporto migliorato della parola chiave LIKE.

Capacità trasversali

Miglioramenti

- Maggiore robustezza per l'API ErrorID
- Ottimizzazione delle prestazioni per la trasformazione di progetti di grandi dimensioni. Ad esempio: timeout per saltare i file bloccati, riutilizzo della classificazione di AWS Transform for mainframe refactor e migliori allocazioni di memoria.
- Ottimizzato l'ingombro della memoria durante la trasformazione COBOL/. PL1
- CVE fisso su terze parti (jQuery e bootstrap).
- Opzioni TimeoutParser gestite in TC.

- È stata migliorata la riscrittura di più spazi sulle query SQL.
- Cursore di sola lettura migliorato con attributo di sensibilità.

Note di rilascio 4.0.0

Data di rilascio: 8 aprile 2024

Per istruzioni su come migrare da AWS Transform for mainframe Runtime 3.10.0 a 4.0.0, vedere. the section called "Migrazione da 3.10.0 a 4.0.0"

Questa versione di AWS Transform for mainframe Runtime and Modernization Tools è incentrata sull'aggiornamento delle dipendenze critiche e delle tecnologie supportate, migliorando al contempo le prestazioni in molteplici funzionalità. Alcune caratteristiche e modifiche chiave di questa versione sono:

- • Esegui l'aggiornamento da Spring Boot 2.7 a 3.2.4, Spring Core 5.3 a 6.1.5 e Tomcat da 9.0 a 10.1.17 per fornire maggiore sicurezza, prestazioni e manutenibilità utilizzando versioni che vengono aggiornate e mantenute attivamente.
  - Caricamento lento sull'applicazione front-end per creare progetti di grandi dimensioni più veloci con più di 2000 schermate e ridurre l'inizializzazione della visualizzazione da 10 s a 300 ms.
  - Supporto per la visualizzazione DBCS sull'applicazione front-end per il miglioramento del supporto dei caratteri a doppio byte per fornire un nuovo carattere che gestisca caratteri a doppio byte e a byte singolo, impedisca l'immissione a byte singolo in un campo a doppio byte e gestisca campi con caratteri misti a doppio byte e a byte singolo.
  - Funzionalità AS400 di AS400 monitoraggio dei thread per l'applicazione online per eseguire applicazioni con parallelizzazione.
  - Prestazioni migliorate in termini di contesto e RunUnit inizializzazione grazie all'aggiunta di un meccanismo configurabile per preinizializzare il contesto del programma, riducendo l'impatto del caricamento di strutture complesse inerenti alla complessità legacy.

Questa versione di AWS Transform for mainframe Runtime è stata testata con lo stack seguente. Potrebbero essere compatibili anche altre versioni.

Componente	Versione testata
Java	Java 17

Livello di presentazione	Nodo JS 18.18
	Npm 9.8
	Angolare 16.1
Livello di servizio	Spring Boot 3.2.4
	Spring Core 6.1.5
	Spring State Machine 4.0.0
Livello di persistenza	Motore PostgreSQL 14
	Oracle 21
Server applicazioni	Apache Tomcat 10.1.17

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

Versione di runtime 4.0.0

zOS

Nuove funzionalità

- È stato aggiunto il supporto per includere l'istruzione «-INC CPYNAME».
- CICS - Aggiunto il supporto per l'istruzione HANDLE. PUSH/POP
- COBOL - Aggiunto il supporto per «ASSIGN TO DYNAMIC».
- È stato aggiunto il supporto per DB2 UNLOAD utilizzando INFUTILB.
- È stato aggiunto il supporto per la parola chiave SEQNUM in un'istruzione OVERLAY di INREC.

Miglioramenti

- SORT - Aggiunto il supporto per caratteri speciali (parentesi e asterischi) nelle stringhe di ordinamento letterali C'... '.
- SORT - Supporto migliorato per l'argomento OUTFIL NOMATCH- (..).
- SORT - Aggiunto il supporto per la definizione dei dati SYMNAMES.

- SORT - Migliore gestione degli argomenti TO= e LENGTH=.
- SORT - Migliore gestione della disposizione MOD.
- SORT - Aggiunto il supporto per l'argomento HIT=NEXT.
- ICEGENER migliorato per aggiungere il supporto per la codifica di file di output specifici.
- INFUTILB - Supporto migliorato per la clausola WITH UR.
- INFUTILB - Supporto migliorato per lo scaricamento quando è falso. writeNullIndicator
- DSNUTILB - Maggiore robustezza nella fase di caricamento quando la parola chiave NULLIF segue una parola chiave SQL opzionale.
- DSNUTILB - Supporto migliorato per isolare il nome delle colonne.
- DSNUTILB - Aggiunto il supporto per caricare un file vuoto in una tabella.
- DNSUTILB - Aggiunto il supporto per la disposizione MOD per il file DNSUTILB SYSDISC.
- IDCAMS - Supporto avanzato per i commenti.
- JCL - Aggiunto il supporto per la colonna con virgolette doppie. LoadTask
- JCL - Gestione avanzata delle query SQL UNLOAD relative alla rimozione dei passi bianchi.
- JCL - Risposta migliorata dello script Groovy quando si verifica un'eccezione durante l'elaborazione per garantire un formato JSON.
- JCL - Migliore disposizione dei file di controllo nel caso di DISP=NEW e DISP=OLD.
- JCL - Supporto migliorato per gestire più riferimenti di generazione GDG con caratteri speciali nel nome base GDG.
- JCL - Supporto avanzato per caricare un file fittizio.
- JCL - Supporto avanzato per tempFilesDirectory il parametro YML.
- JCL - Restituzione JSON migliorata quando è necessario evitare le virgolette doppie all'interno di un elemento stringa.
- JCL - Migliorato FileUtils per supportare il nome base GDG.
- JCL - Programma DSNTEP avanzato per l'esecuzione di più query. DB2
- Aggiunto il supporto per Spring beans.
- Migliorato SQLConverter per evitare di correggere date errate.
- Migliore JicsTimeBuilder gestione di YYYYDDD.
- È consentito l'accesso ai barattoli personalizzati da Groovy.

- IMS - Navigazione migliorata tra i record nell'implementazione del database IMS.
- IMS - CBLTDLI migliorato per poter avviare il programma use purge.
- IMS: DFSRRC00 in grado di passare i parametri dal programma groovy al programma di backend.
- Aggiunto il supporto per il comando JICS che non è stato richiamato tramite TransactionRunner.
- JICS: prestazioni migliorate utilizzando la cache configurabile.
- <noloc>Blusam</noloc> - Aggiunge il supporto per disabilitare il riscaldamento <noloc>Blusam</noloc> all'apertura per migliorare le prestazioni di set di dati di grandi dimensioni.
- <noloc>Blusam - Comportamento migliorato sui normali set di dati Blusam</noloc><noloc>. delete/rename </noloc>
- <noloc>Blusam</noloc> - Prestazioni migliorate nelle operazioni di registrazione.
- Semplificatore di dati migliorato per i metodi che determinano se una stringa ha un valore basso.
- Supporto migliorato per il problema dei decimali compressi e del tipo di ordinamento.
- Configurazione avanzata DB2 come fonte di dati principale con AWS Secrets.
- FileSystem API migliorata per esporre lo stato del file.
- Input migliorato del flusso di DynamicFileBuilder lettura con LineSeparator.
- Semplificatore di dati migliorato per i metodi che determinano se una stringa ha un valore basso quando si tratta di charset. CUSTOM930
- SQL: elaborazione dell'output di SQL Stored Procedure migliorata.
- SQL - Mappatura lambda migliorata per più tabelle con alias.
- COBOL - Supporto migliorato per l'istruzione LENGTH OF.
- COBOL - Aggiunto il supporto per l'istruzione TRANSFORM.
- COBOL - Aggiunto il supporto per 9 nuove funzioni matematiche.
- COBOL - Supporto migliorato per FUNCTION. INTEGER-OF-DAY
- COBOL - Supporto migliorato per 88 livelli che coinvolgono il valore figurativo.
- COBOL - Trasformazione migliorata per l'istruzione SET ADDRESS.

AS400

Nuove funzionalità

- Sono state rimosse le entità indicatrici duplicate.

- È stato aggiunto il supporto per i caratteri DBCS.
- È stata introdotta la gestione della parola chiave HELP per il controllo dei record dei sottofile.
- È stato aggiunto un parametro di configurazione per attivare o disattivare la maiuscola del nome delle colonne e dividere il contenuto delle colonne di commento su pipe char.
- È stato aggiunto il supporto per l'utilizzo di 0x0c come ultimo nibble per i campi di tipo Packed.
- RPG - Prototipi gestiti dichiarati con ('system'). ExtProc
- CL - Il parametro 'CLEAR' gestito del comando cl-RMVMSG + introduce code di messaggi non programmabili in memoria.
- CL - Ha gestito il passaggio di istruzioni generiche alle chiamate SBMJOB CMD ().
- CL - Aggiunti i comandi STRCMTCTL e ENDCMTCTL. Meccanismo di blocco modificato e pulizia delle transazioni e dei blocchi.
- CL - Aggiunto il supporto per il parametro RCDDLM per il comando CPYTOIMPF.
- CL - Aggiunta la gestione degli zeri di riempimento nel comando SAVOBJ.
- CL - Aggiunta la gestione delle librerie incluse nel nome qualificato del parametro OBJ per RTVOBJD.
- CL - Aggiunto il supporto per i parametri di comando CPYTOIMPF STRDLM, STRESCCHR e RMVBLANK.
- CL - CPF2407 CPF2419 RTVMGS file/id migliorato da inviare e quando non viene trovato.
- CL - Comando RCVF migliorato per ricevere record da qualsiasi libreria fornita nel parametro DEV.

Miglioramenti

- Valori predefiniti modificati per l'esecutore di attività Blu4iv per consentire una migliore scalabilità per impostazione predefinita.
- Parameterhelper modificato per convertire l'elenco di stringhe e in String. ElementaryRangeReference
- CTID migliorato per gestire colonne non esistenti in POSTGRE.
- Maggiore robustezza per supportare l'API dello spazio utente «QUSPTRUS».
- È stato aggiunto il supporto per gli spazi utente QUSRUSAT e QUSCUSAT. APIs
- Supporto migliorato per User Space API (QUSPTRUS) senza codice di errore.
- È stato aggiunto il supporto per CRON Job Scheduling utilizzando Quartz.

- Supporto migliorato del ciclo del programma RPG.
- Migliore gestione delle transazioni Blu4iv.
- È stato migliorato il blocco dei record dei file sottoposti al controllo dell'impegno all'interno della stessa transazione.
- Migliore gestione dell'inizializzazione dei sottofile.
- Visualizzazione migliorata degli indicatori di scorrimento per le righe dei messaggi.
- Sono stati evitati gli zeri finali sui numeri inviati tramite la coda di dati.
- Schermata di informazioni aggiuntive sui messaggi migliorata.
- Operazioni di scrittura JPA migliorate per considerare la libreria corrente.
- Comportamento migliorato ProgramJobExecutor durante l'esecuzione di programmi senza parametri.
- Aggiunta funzionalità per passare direttamente gli argomenti dai link front-end agli script di back-end.
- Migliore gestione delle transazioni per i metadati dei lavori.
- CL - Aggiunto il supporto per il parametro SECLVL in RTVMSG.
- CL - Aggiunta un'implementazione vuota per CLRLIB.
- CL - Supporto CPYFRMIMPF migliorato per la copia da database e CSV.
- CL - Implementazione migliorata del CPYFRMIMPF per ignorare le colonne aggiuntive.
- CL - Interpretazione migliorata in CPYTOIMPF e CPYFRMIMPF dei parametri di formattazione precedenti.
- CL - Aggiunto un parametro per formattare i valori numerici in SAVOBJ. removeDecimalPoint
- CL - Comando RCVF migliorato per gestire correttamente la condizione EOF.
- CL - RTVSYSVAL - Implementazione SYSVAL = QDATETIME.
- CL - Comando OVRDBF modificato per ottenere il campo come nome di tabella predefinito.
- CL - RTVJOBA Valore non disponibile per il parametro: USRLIBL.
- CL - Ha gestito le barre iniziali nel parametro SNDPGMMSG MSGF.
- CL - Supporto migliorato per i caratteri jolly nel file sorgente nel comando DSPFFD.
- CL - Migliore gestione del parametro PGMQ in RCVMSG e SNDPGMMSG.
- CL - Il parametro RTVMSG è stato reso opzionale per l'allineamento con i documenti precedenti.

Funzionalità trasversali

Nuove funzionalità

- Funzionalità migliorata durante il passaggio del parametro alla clausola USING del cursore OPEN.
- Prestazioni: preinizializzazione migliorata del contesto e RunUnit ottimizzazione delle prestazioni.

Miglioramenti

- È stato migliorato il meccanismo per scaricare i valori bassi dal comando UNLOAD del programma di utilità INFUTILB.
- Aggiunta l'opzione di supporto dello schema corrente sul gestore segreto di datasources.
- Runtime migliorato per non considerare i parametri passati al cursore aperto quando non sono necessari.
- Convalida del formato numerico migliorata per i campi numerici.
- Diagnostica SQL migliorata in un ambiente di esecuzione altamente parallela.
- È stato introdotto l'unicode per la sequenza di byte della codepage (FE FD).
- DataSimplifier ottimizzazione delle prestazioni: istruzioni di assegnazione migliorate.
- DataSimplifier ottimizzazione delle prestazioni: migliora il valore predefinito per l'inizializzazione dei tipi numerici per evitare utilizzi inutili. BigDecimal

Versione 4.0.0 degli strumenti di modernizzazione

zOS

Nuove funzionalità

- Aggiunto il supporto per la gestione del programma Abend.
- Supporto migliorato per generare set di dati AIX.
- COBOL - Aggiunto il supporto per la clausola JUSTICIED sui campi. ALPHANUMERIC/ ALPHABETIC/GRAPHIC

Miglioramenti

- Migliore gestione degli attributi PURGETHRESH per le definizioni delle risorse TRANSCLASS.
- Supporto migliorato per la definizione dei dati e l'istruzione MOVE.

- CICS - Supporto migliorato per il comando DELAY sull'opzione MILLISECS.
- Mappatura lambda SQL migliorata per più tabelle con alias.
- Supporto migliorato per la ricerca dei campi principali.
- Set SQLCA sqlstate migliorato per le operazioni COMMIT e ROLLBACK.
- COBOL: migliora l'analisi commentando paragrafi obsoleti
- COBOL - Supporto migliorato per la clausola REPLACING.
- COBOL - Aggiunto il supporto per le funzioni matematiche ASIN ACOS LOG TAN.
- COBOL - Aggiunto il supporto per più istruzioni AFTER in PERFORM VARIYING.
- COBOL - Supporto migliorato per i campi RENAMES (livello 66).
- COBOL - Metodo LENGTH OF avanzato per ottenere la lunghezza in base a un indice specifico in un campo array.
- COBOL - Aggiunto il supporto per più clausole AFTER nelle istruzioni PERFORM VARYING.
- COBOL - Supporto migliorato per la clausola RENAMES.
- COBOL - Supporto migliorato della parola chiave PICTURE.
- COBOL - Supporto migliorato per l'analisi dei campi di livello 88.
- COBOL - Condizione goto dipendente migliorata con gli elementi di dati della tabella.

AS400

Nuove funzionalità

- Aggiunta funzionalità per passare argomenti alle chiamate java front-end dirette.
- CL - Generazione %SST migliorata, incluso il supporto per *LDA con CL→Java.
- RPG - Aggiunto il record Program-Descripted per i file DISK.

Miglioramenti

- File di visualizzazione migliorato, risolvi i campi di riferimento con la parola chiave «REFFLD».
- Supporto migliorato per la parola chiave del file di visualizzazione SETOF-CSRLOC.
- File rimossi dal controllo di impegno dopo la chiusura.
- Comportamento coerente garantito per le operazioni di lettura e scrittura simultanee su una tabella eseguite dallo stesso programma.

- Assegnazione gestita alla sottostringa di. SizePrefixedAlphanumericType
- Gestito il passaggio della struttura dei dati alla procedura con un parametro di stringa di lunghezza variabile.
- Migliore conservazione dei valori numerici non validi durante l'evento OnBlur e creazione di listener di eventi solo per campi validi.
- Messaggi di errore migliorati sugli schermi ed evidenziazione dei campi con input non valido.
- Migliore gestione dei campi dello schermo condizionati dagli indicatori.
- Scorrimento abilitato con la rotellina del mouse.
- Aggiunto il supporto per i tasti funzione per la schermata di aiuto.
- Supporto migliorato per il testo lungo nel split-dynamic-field componente.
- Migliore gestione dei file LF multi-record durante la ridenominazione dei record.
- CL - Comando RTVJOBD migliorato per gestire i file LF (visualizzazioni).
- CL - Comando OVRDBF migliorato se usato su un LF con più record.
- RPG - Scenario gestito in cui la procedura definisce una variabile con lo stesso nome del parametro rinominato.
- RPG - Migliore gestione di*ZEROS durante l'inizializzazione di BinaryInteger firmato.
- RPG - Migliore gestione dei puntatori a variabili non locali (di riferimento).
- RPG - Migliore gestione delle istruzioni ELSEIF successive alle istruzioni. IFxx
- RPG - Aggiunto il supporto per i campi definiti con LIKE sul prototipo.
- RPG - Migliorato il supporto per la parola chiave LIKE di un campo creato da LIKEREC.
- RPG - Migliore generazione di operatori con figurativi.
- RPG - Analisi migliorata per l'espressione di array xxx (\ *) e supporto in %lookup.
- RPG - Codice LookUp operativo migliorato con indicatori alto e uguale (o basso e uguale).
- RPG - Analisi dei formati liberi migliorata.
- RPG - Analisi migliorata delle costanti denominate i-Card che seguono i formati di registrazione i-Card.
- RPG - Supporto migliorato per i tipi INTEGER e UNSIGNED.
- COBOL - Aggiunta la clausola INDIC di supporto del formato DSPF nell'istruzione COPY DDS.
- COBOL - Grammatica migliorata per le istruzioni DISPLAY e ACCEPT per sbloccare la trasformazione e la generazione.
- COBOL - Aggiunto il supporto per i file DISK.

- COBOL - Programmi di supporto migliorati per i file di visualizzazione DDS.
- COBOL - Aggiunto il supporto per la clausola LIKE.
- COBOL - Aggiunto il supporto per il file DISK descritto dal programma.
- COBOL - Aggiunto il supporto per il nome di file con suffisso.

Funzionalità trasversali

Nuove funzionalità

- Ha gestito il caricamento lento dei componenti cartografici dei progetti web.

Miglioramenti

- Migliore generazione in Java dei parametri degli indicatori SQL.
- Migliore capacità di gestire le variabili coinvolte nell' DB2 istruzione SET.
- Migliore aumento dell'errore alla fine del cursore recuperato quando l'output è un array a singola entità.
- Percorso gestito in Linux.
- Data Migrator gestisce le vulnerabilità e rimuove le dipendenze inutilizzate.

Note di rilascio 3.10.0

Questa versione di AWS Transform for mainframe Runtime and Modernization Tools si concentra sugli aggiornamenti e sui miglioramenti di base di base del prodotto, con l'obiettivo di aumentare le prestazioni e la robustezza in tutte le fasi di trasformazione ed esecuzione. Alcune funzionalità 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 <noloc>Blusam</noloc> 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

- <noloc>Blusam</noloc> - 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
- <noloc>Blusam</noloc> - Aggiunto il supporto per la cancellazione di Ehcache dei set di dati <noloc>Blusam tramite</noloc> un nuovo metodo denominato .removeCache () CoreBluesamManager
- <noloc><noloc>Blusam - Comportamento migliorato per i normali set di dati Blusam delete/rename </noloc></noloc>
- 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 rdw/recordSize lettura dal catalogo durante la creazione del set di dati

- JCL - Aggiunto il supporto per aggiornare l'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 <noloc>Blusam</noloc> 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 relativi APIs
- 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 di 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 help

- 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 DataUtils di confronto. compareAlphInt() 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, -JAVA-ORGSPRINGFRAMEWORKSECURITY-5905484, CVE-2023-46120, CVE-2023-6481, CVE-2023-6481, CVE-2023-6378, CVE-2023-5072) IN1

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

- Inserimento migliorato dei parametri del cursore per il valore low/high
- 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 IMPORT/EXPORT parole chiave
- 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 - È stata 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 Input/Output
- DSPF - Migliore gestione dei separatori di raggruppamento durante la convalida del campo numerico
- Record di mappatura output/DDS migliorati
- 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 entro le condizioni IF/ELSEIF/WHEN
- RPG - Migliore gestione del comando READ quando viene chiamato su un file dspf
- RPG - Supporto migliorato 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 Transform for mainframe 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 funzionalità 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 in AS400, con la caratteristica principale che le aziende possono inviare messaggi di richiesta in modo sincrono tra di 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 dei 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 .yml 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: leggi gli attributi dal catalogo durante la creazione del set di dati rdw/recordSize

- 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 dei blocchi aperti 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 <noloc>blocco di Blusam</noloc>

AS400

Nuove funzionalità

- Crea un registro nel 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 COBOL AS400
- Gestisce il comando cl RMVJOBSCDE
- Miglioramento per RUNSQL/DLYJOB
- CHKOBJ: generazione del codice di errore precedente per il parametro LIB
- SNDPGMMSG: 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
- SNDPGMMSG e RCVMSG: sono supportati parametri aggiuntivi MSGF, MSGFLIB, MSGDTA, MSGTYPE, KEYVAR, MSGKEY, MSGID

Miglioramenti

- Le schede I/O WORKSTATION migliorate supportano
- 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 DAOs 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 legate ai puntatori bad/invalid
- Migliorato MessageHandlingBuilder per visualizzare msgId e MsgFile name come stringhe per RCVMSG
- 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 DEV/RCDFMT parametri 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 YAML database.cursor.raise.already.opened.error per consentire al framework di generare l'errore SQLCODE 502 quando il cursore già aperto si apre

Miglioramenti

- Aggiunti gapwalk poms a AWS Transform for mainframe on 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 delle single/double virgolette nelle stringhe letterali

AS400

Miglioramenti

- Gestione migliorata della funzione integrata %editc del codice di modifica X 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
- È stato migliorato il filtro delle parole chiave propagate tramite la parola chiave ref/reffield
- Struttura dati senza nome DataArea 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
- Test case 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
- Supporti per schede IMPROVE WORKSTATION I/O

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 dell'API comune per registrare errori e/o informazioni in COBOL2 Model, RPGCycle Builder, 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 Transform for mainframe 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.

- Novità APIs disponibili per operare e interagire meglio con i set di dati <noloc>Blusam</noloc>, 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 comune: gestione delle dimensioni dei record nei dati in-stream DDUtils
- File indicizzato: gestione dell'opzione GENKEY

Miglioramenti

- Servizi di caricamento <noloc>Blusam esternalizzati in un jar separato</noloc>
- 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
- <noloc>Blusam</noloc>: Evita il riscaldamento duplicato quando si utilizzano più istanze
- Ingombro di memoria ridotto 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 I/O schede 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 in INFDS ROW/COL
- Supporto migliorato 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
- Aggiunto pagination/filtering all'elenco degli 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 <noloc>Blusam</noloc> 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 applicare la codifica e la decodifica codeshiftpoint su database mirati 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 with 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 Â¥ nelle variabili
- JCL: calcolo di RC in base ai passaggi precedenti
- Confronto tra byte anziché stringhe quando si usa SUBSTR PL1
- 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

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

Miglioramenti

- Supporto migliorato per il condizionamento Control Level
- Migliore restituzione dei prototipi con LIKE
- Supporto migliorato per la gestione delle funzioni %mesi, %anno, %giorni
- Support 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 LIKEREC
- 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 Transform for mainframe 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, cache <noloc>Blusam</noloc> e blocchi.
- Endpoint di monitoraggio che consente di recuperare i parametri relativi a transazioni, 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 nel DB2 formato 24:00:00 ore (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 AS400 gli schermi (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)

- Support per la scrittura nel logger anziché nel flusso di output di sistema predefinito nei file SYSPRINT e SYSPUNCH predefiniti (V7-10098)
- Supporta la cache <noloc>Blusam</noloc> 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)
- Support 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 REFSHIT DSPF (V7-10049)

Miglioramenti

- Miglioramento nella gestione della parola chiave di descrizione del file EXTIND (*) (V7-7404) INUx
- Migliore trasformazione dei file SQLDDS (V7-7687)

- Gli oggetti di file non sono più generati per i file (V7-9062) AS400
- 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)
- Support per il campo inizializzato su*USER (V7-9806)
- Support per digitare COMP AS400 (V7-9840)
- COBOL400 Analisi 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)
- Support per chiamate MQ in RPG (V7-10007)
- %LOOKUP integrato migliorato (struttura dati ad array con chiavi) (V7-10022)
- Support 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)
- Support 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)
- Migliorato GOTOs con la condizione Else (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 Transform for mainframe Runtime and Modernization Tools offre nuove funzionalità sia per gli zOS che per le migrazioni AS400 precedenti, 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 principali modifiche apportate in 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 AWS Transform for mainframe 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 DB2 funzioni SQL integrate quando si chiama l'utilità INFUTILB UNLOAD (V7-9554)
- Assegnazioni avanzate di array multidimensionali PLI (V7-9592)
- Gestione del reindirizzamento sysout al file (V7-9992)

Miglioramenti

- Aggiunge l'attivazione di procedure memorizzate per RDBMS (V7-9155) DB2
- 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 handle different termination/dispositionin set di dati concatenati (V7-9653)

- Rendi PageSize configurabile per i set di dati Blusam (<noloc>V7-9680</noloc>)
- DSNUTIL: consente il caricamento delle 24:00:00 come ORA valida in LUW (V7-9697) DB2
- Supporta il confronto HIGH-VALUES (0xff) in NumberUtils .ne () NumberUtils /.eq () (V7-9731)
- JCL/GROOVY - supporta DO... Parole chiave THEN nelle IF-THEN-ELSE clausole IDCAMS per eseguire più istruzioni in un singolo blocco (V7-9750)
- Programma chiamato JHDB non valido esterno a Runner (V7-9782) JHDBBatch
- 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)
- primary/secondary File di supporto (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 il controllo degli enable/disable impegni (V7-9956)
- Supporta gli indicatori esterni sulle parole chiave COMMIT

Miglioramenti

- Migliora la visualizzazione dei valori 0 e l'EDTWRD (V7-8933)
- Support 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)
- <noloc>Migliora la registrazione per Jics e Blusam (V7-8847)</noloc>
- 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)

- <noloc>Migliora la concorrenza di scrittura dei thread per il registro Blusam Locks (V7-9505)</noloc>
- Supporta la configurazione di più origini dati per Utility-PGM (V7-9570)
- <noloc>Modalità</noloc> 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 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 (V7-9809) DB2
- CICS - Migliora il comando QUERY SECURITY per l'opzione LOGMESSAGE (V7-9969)
- PL1 - Migliora la generazione di sacchetti integrati (V7-9989) CHARG/chargraphic

Miglioramenti

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

- Miglioramento della risoluzione della variabile PL1 host 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)
- Generazione di file Improve and AllIndexes (V7-9818) SQLModel
- 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 Transform for mainframe Runtime and Modernization Tools offre nuove funzionalità sia per gli zOS che per le migrazioni AS400 precedenti, 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, GET/PUT supporto esteso e recupero automatico dei metadati della coda.
- REST Endpoint for dataset, 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
- AS400
- 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)
- Porto di To & (V7-9212) RedisJics TSQueue 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 l'endpoint <noloc>get</noloc> dataset per l'API Blusam
- JCL - AGGIUNGI il supporto per il lavoro in batch con nome che include caratteri speciali come # (V7-9136)
- TSModel il recupero 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 delle ORD/UNORD abbreviazioni (V7-9451)
- Miglioramenti delle prestazioni del <noloc>blocco Blusam Redis (V7-8639</noloc>)
- Migliora l'istruzione DEFINE CLUSTER per supportare RECORDSIZE fornito nell'ambito degli argomenti DATA () (V7-9337)
- Aggiunge il supporto degli BUFFERSPACE/UNIQUE attributi nelle istruzioni DEFINE CLUSTER (V7-9419)
- Migliora l'operazione di lettura di <noloc>Blusam</noloc> 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)
- <noloc>Supporto per caratteri non stampabili nel nome Blusam TS Queues (V7-9212)</noloc>
- Gestisci la ricezione della creazione di mappe per il campo con mapset null (V7-9486)
- Migliora le operazioni di BluesamRelativeFile cancellazione e riscrittura per la modalità di accesso dinamico. (V7-8989)

AS400

Nuove funzionalità

- Aggiungi una funzionalità per generare file CL come programmi Java tramite DS/STM pivot standard (V7-9427)
- Supporta file di input con modalità ADD (V7-9378)
- Migliore gestione dell'ordinamento e del recupero per supportare il comando cl OPNQRYF (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. ResulSet (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 PL1 GRAFICA 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 del supporto di 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 (V7-9352) AND/OR
- 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, tipo (V7-9542) Date/Time/Timestamp

- Includi SQLCA dall'analisi (V7-9333) FullFree
- 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 Â¬ carattere con! nel caso in cui Â¬ 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 Trasforma per le vulnerabilità di sicurezza del mainframe

Common Vulnerabilities and Exposures (CVE) è un elenco di riferimento per le vulnerabilità di sicurezza informatica note pubblicamente. Ogni voce contiene un numero di identificazione, una descrizione e almeno un riferimento pubblico.

Ti consigliamo di eseguire sempre l'aggiornamento alla versione più recente di AWS Transform for mainframe per proteggerti dalle vulnerabilità note. Le scansioni di sicurezza vengono eseguite continuamente con Amazon Inspector e i risultati vengono classificati in base alla loro gravità nel NIST.

L'elenco seguente descrive in dettaglio le CVEs correzioni presenti in ciascuna versione secondaria disponibile, derivanti dall'uso delle dipendenze:

Versione	CVE
5.75.0	CVE-2026-22610, CVE-2025-67735, CVE-2025-66035, CVE-2025-11226, CVE-2025-64718
5.1.0	CVE-2025-55754, CVE-2025-55752, CVE-2025-61795, CVE-2020-36843, CVE-2025-11226, CVE-2025-41254, CVE-2025-41248, CVE-2025-59052, CVE-2025-41249, CVE-2025-4949, CVE-2024-8184, CVE-2024-6763, CVE-2025-58057, CVE-2025-58056
4,10,0	CVE-2025-55163, CVE-2025-48924, CVE-2025-53864, CVE-2025-52999, CVE-2025-41234, CVE-2025-49125, CVE-2025-48988, CVE-2025-5889
4.9.0	CVE-2025-49146, CVE-2025-22871, CVE-2022-41404, CVE-2025-22233, CVE-2025-46701, CVE-2025-22235, CVE-2025-46392, CVE-2025-27820, CVE-2025-24813

Versione	CVE
4.8.0	CVE-2025-22228
4.7.0	CVE-2025-25193, CVE-2025-24970, CVE-2024-47535, CVE-2024-38820, CVE-2024-38809, CVE-2024-22262, CVE-2022-45868
4.6.0	CVE-2024-12801, CVE-2024-12798, CVE-2024-50379, CVE-2024-56337
4.5.0	CVE-2024-47535, CVE-2024-52316, CVE-2024-47535, CVE-2024-38827
4.4.0	CVE-2024-38820, CVE-2024-38821, CVE-2024-38809, CVE-2024-38816, CVE-2024-47554, CVE-2024-6484, CVE-2024-6485
4.3.0	CVE-2024-43788, CVE-2022-25898, CVE-2021-30246, CVE-2024-21484, CVE-2024-34750
4.2.0	CVE-2020-11023, CVE-2023-26364, CVE-2019-11358, CVE-2020-11022, CVE-2021-23358, CVE-2017-18214, CVE-2022-24785, CVE-2022-31129, CVE-2023-48631
4.1.0	CVE-2024-29025, CVE-2024-23080, CVE-2024-22262, CVE-2024-30171, CVE-2024-29857, CVE-2024-30172
4,0,0	CVE-2016-1000027, CVE-2022-1471, CVE-2024-1597, CVE-2024-22243, CVE-2024-22233, CVE-2024-22234, CVE-2024-22259, CVE-2024-22257, CVE-2024-29131, CVE-2024-29133

> **Note**
>
> Per i dettagli sui CVEs problemi risolti nelle versioni precedenti, contatta il tuo responsabile della distribuzione di AWS Transform for mainframe

Istruzioni per l'aggiornamento di AWS Transform for mainframe

Questa pagina contiene le istruzioni per l'aggiornamento della versione AWS Transform for mainframe.

Argomenti

- Aggiornamenti comuni
- Migrazione da 3.10.0 a 4.0.0
- Migrazione da 5.85.0 a 5.86.0

Aggiornamenti comuni

Nella maggior parte dei casi, quando si aggiorna la versione AWS Transform for mainframe Runtime, è necessario sostituire gli elementi (file di configurazioneWARs, script, ecc.) della versione precedente con quelli forniti nella nuova versione e riavviare l'applicazione. Assicuratevi di eseguire test di regressione completi delle applicazioni modernizzate dopo l'aggiornamento. Puoi anche contattare il responsabile della distribuzione di AWS Transform for mainframe per istruzioni specifiche applicabili alla tua applicazione.

Alcuni aggiornamenti potrebbero richiedere una configurazione aggiuntiva per garantire la compatibilità. In tal caso, segui le istruzioni per l'aggiornamento specifico.

Migrazione da 3.10.0 a 4.0.0

La modifica principale in 4.0.0 è la migrazione da Spring Boot 2.7 a Spring Boot 3.2 e da Tomcat 9 a Tomcat 10.

Modifiche al codice

Questa sezione elenca le modifiche necessarie per rendere il codice modernizzato compatibile con AWS Transform for mainframe Runtime 4.0.0. Puoi saltare questa sezione se decidi di lanciare

una nuova generazione utilizzando la versione 4.0.0 su AWS Transform for mainframe refactor (Transformation Center).

Modifiche al POM

Gruppo	ArtifactId	Modifica
org.slf4j	slf4j-api	Rimuovi (è una dipendenza transitiva)
org.yaml	snakeyaml	Rimuovi (è una dipendenza transitiva)
org.springframework.boot	spring-boot-starter-web	- Aggiorna spring.boot.versio n alla 3.2.4 - Rimuovi l'esclusi one di log4 4j j-to-slf
org.springframework.boot	spring-boot-starter-jta-ato mikos	Passa a com.atomikos: 3-starter:6.0.0 transactions-sprin g-boot
org.apache.commons	commons-dbcp2	Aggiornamento alla versione 2.10.0
org.postgresql	postgreql	Aggiornamento a 42.7.2
com.microsoft.sqlserver	mssql-jdbc	Aggiornamento a 12.4.2.jre11
com.oracle.database.jdbc	ojdbc8	Passare alla versione ojdbc11 23.3.0.23.09

Migra da Javax a Giacarta

L'aggiornamento tomcat include una migrazione dal pacchetto Javax Java a Jakarta. Assicurati di aggiornare di conseguenza le importazioni da javax.* a jakarta. *.

Quasi tutte le vecchie classi referenziate nel pacchetto Javax si trovano a Giacarta. Le eccezioni note a questo sono i `javax.xml` pacchetti `javax.sql` and, che sono ancora invariati.

Atomikos cambia

A causa della modifica della dipendenza di cui sopra, i riferimenti a `org.springframework.boot.jta.atomikos.AtomikosDataSourceBean` devono essere modificati in. `com.atomikos.spring.AtomikosDataSourceBean`

Rimozione del dialetto PostgreSQL

La classe personalizzata viene rimossa. `PostgreSQLDialect.java` Anche i riferimenti ad essa nel programma di avvio principale devono essere rimossi.

Distribuzione (AWS Transform for mainframe Runtime)

Tomcat

Questa versione è compatibile con `10.1.17` Tomcat. L'aggiornamento del server Tomcat a questa versione è necessario per eseguire il Runtime. AWS Transform for mainframe `4.0.0` Assicurati di eseguire il porting delle vecchie modifiche alla configurazione (in particolare le proprietà di Catalina).

Dipendenze condivise

La cartella condivisa di runtime contiene le up-to-date dipendenze.

Dipendenze aggiuntive

Se sono state utilizzate dipendenze aggiuntive (non incluse nel runtime), potrebbe essere necessario aggiornarle. Il file readme nella cartella aggiuntiva elenca le versioni supportate.

Migrazione da 5.85.0 a 5.86.0

La modifica principale in 5.86.0 (Versioni preliminari Alpha) è la migrazione da Spring Boot 3.5.7 a Spring Boot 4.0.3 e da Tomcat 10 a Tomcat 11.

Modifiche al codice

Questa sezione elenca le modifiche necessarie per rendere il codice modernizzato compatibile con AWS Transform for mainframe Runtime 5.86.0. Puoi saltare questa sezione se decidi di lanciare una nuova generazione utilizzando la versione 5.86.0 su Blu Insights (Transformation Center).

Modifiche al POM

Aggiorna la versione Spring Boot:

```
<properties>
    <spring.boot.version>4.0.3</spring.boot.version>
```

```
</properties>
```

Sostituisci `spring-boot-starter-web` con`spring-boot-starter-webmvc`:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-webmvc</artifactId>
    <scope>provided</scope>
</dependency>
```

Riorganizzazione dei pacchetti

A causa della modifica delle dipendenze di cui sopra, i riferimenti a:

- `org.springframework.boot.autoconfigure.domain.EntityScan`deve essere cambiato in. `org.springframework.boot.persistence.autoconfigure.EntityScan`
- `org.springframework.boot.autoconfigure.jdbc.DataSourcePropertiesdeve` essere cambiato in`org.springframework.boot.jdbc.autoconfigure.DataSourceProperties`.

Distribuzione (AWS Transform for mainframe Runtime)

Tomcat

Questa versione è compatibile con `11.0.15` Tomcat. L'aggiornamento del server Tomcat a questa versione è necessario per eseguire il Runtime. AWS Transform for mainframe `5.86.0` Assicurati di eseguire il porting delle vecchie modifiche alla configurazione (in particolare le proprietà di Catalina).

Dipendenze condivise

La cartella condivisa di runtime contiene le up-to-date dipendenze.

Dipendenze aggiuntive

Se sono state utilizzate dipendenze aggiuntive (non incluse nel runtime), potrebbe essere necessario aggiornarle. Il file readme nella cartella aggiuntiva elenca le versioni supportate.

AWS Trasformazione per il ciclo di vita del mainframe

Questa sezione definisce le date di fine del ciclo di vita (EOL) per le versioni principali di AWS Transform for mainframe Runtime. Ciò consente di pianificare gli aggiornamenti delle versioni in

modo da rimanere aggiornati con le funzionalità e le manutenzione più recenti. Per aggiornare la tua versione, consultathe section called "Aggiornamento di Transform per mainframe AWS ".

Ti consigliamo di verificare la disponibilità di nuove versioni ogni 3 mesi e di eseguire frequentemente l'aggiornamento alle versioni più recenti. Per ogni aggiornamento, è necessario eseguire test non di regressione delle applicazioni modernizzate prima della produzione o delle implementazioni critiche.

> **Note**
>
> Le date di fine del ciclo di vita possono essere soggette a modifiche a causa di importanti misure correttive di sicurezza. Per ulteriori dettagli, consultare Ciclo di vita dei componenti.

AWS Transform for mainframe Runtime End of Life (EOL)

La tabella seguente riassume la data EOL per ogni versione principale.

Versione principale	Data di fine del ciclo di vita
Versione 5	N/D
Versione 4	18 febbraio 2026
Versione 3	8 luglio 2024

> **Note**
>
> La data di fine del ciclo di vita per la versione principale 5 sarà allineata alla data di disponibilità della prossima versione principale.

Per comprendere il modello di supporto delle versioni secondarie, consulta. Ciclo di vita dei componenti

AWS Concetti di Transform for mainframe Runtime

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

Argomenti

- AWS Architettura di alto livello Transform for mainframe Runtime
- AWS Trasformazione per la struttura mainframe di un'applicazione modernizzata
- Cosa sono i semplificatori di dati in AWS Transform for mainframe
- AWS Transform per mainframe Blusam
- AWS Console di Blusam amministrazione Transform for mainframe

AWS Architettura di alto livello Transform for mainframe Runtime

Come parte della soluzione AWS Transform for mainframe per modernizzare i programmi legacy in Java, AWS Transform for mainframe Runtime fornisce un punto di ingresso unificato 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 Transform for mainframe 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 Transform for mainframe 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, fluent APIs.

Argomenti

- AWS Componenti di runtime Transform for mainframe
- Ambienti di esecuzione
- Apolidia e gestione delle sessioni
- Elevata disponibilità e apolidia

AWS Componenti di runtime Transform for mainframe

L'ambiente AWS Transform for mainframe 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 Transform per le librerie mainframe

Le librerie AWS Transform for mainframe sono un insieme di file jar memorizzati in una `shared/` sottocartella aggiunta al classpath standard di tomcat, in modo da renderli disponibili per 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, basate su classi, fluenti). APIs Un esempio importante è la libreria Data Simplifier, che fornisce costrutti di layout e manipolazione della memoria legacy (presenti nei linguaggi COBOL o RPG) ai programmi Java. PL1 Questi jar sono una dipendenza fondamentale del codice Java modernizzato generato dai programmi legacy. Per ulteriori informazioni su Data Simplifier, vedere. Cosa sono i semplificatori di dati in AWS Transform for mainframe

Applicazione Web

Gli archivi delle applicazioni Web (WARs) sono un modo standard per distribuire codice e applicazioni sul server delle applicazioni tomcat. Quelli forniti come parte del runtime AWS Transform for mainframe 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 Transform for mainframe Runtime APIs.

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 importanti categorie di tali categorie 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 Trasformazione per la struttura mainframe di un'applicazione modernizzata All'avvio, AWS Transform for mainframe 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 che si incontrano di frequente:

- 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 Transform for mainframe 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 istruzioni per la riproduzione fluente di Java APIs , `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 (tramite il set di librerie Blusam) e GDG Data Set,
- accesso agli archivi di dati, come RDBMS (dichiarazioni). `EXEC SQL`

Apolidia e gestione delle sessioni

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

La base 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 nel contesto dell'utente una cache to/from esterna (archivio dati).

La gestione dei cookie viene eseguita automaticamente dal framework AWS Transform for mainframe e dal server tomcat sottostante, in modo trasparente per l'utente. Il browser Internet dell'utente gestirà questa operazione automaticamente.

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

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

Elevata disponibilità e apolidia

Più in generale, uno dei principi di progettazione del framework AWS Transform for mainframe 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 Trasformazione per la struttura mainframe di un'applicazione modernizzata

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

- navigazione fluida nelle 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 Transform for mainframe: mappature dei nomi completamente qualificate per recuperare i nomi completi corrispondenti per gli elementi di runtime AWS Transform for mainframe 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 di esempio. COBOL/CICS CardDemo

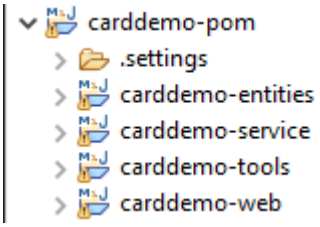
Argomenti

- Organizzazione degli artefatti
- Esecuzione e chiamata di programmi
- Scrivi il tuo programma
- Mappature dei nomi complete

Organizzazione degli artefatti

AWS Le applicazioni modernizzate Transform for mainframe 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 AWS Transform for mainframe Runtime, 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.



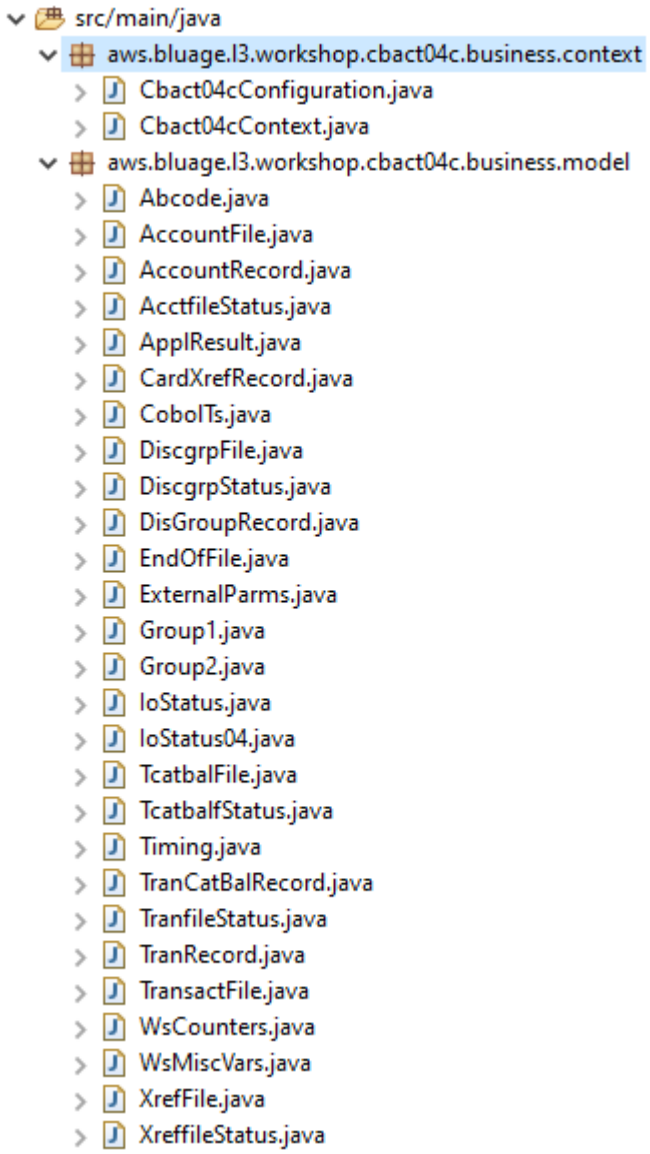
L'organizzazione di base si basa sulla seguente struttura:

- 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 I/O sezione (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 ai risultati COBOL e di PL/I modernizzazione. 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, spiegata nei corsi di formazione di AWS Transform for mainframe. Per ulteriori informazioni, consulta 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 regolarmente il refactoring java, direttamente sul codice sorgente generato, utilizzando qualsiasi approccio di refactoring java che si desideri applicare, a proprio rischio.



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'`@org.springframework.context.annotation.Configuration`annotazione e contiene un singolo metodo che deve restituire un `Configuration` oggetto configurato correttamente.

```
Cbact04cConfiguration.java
1  package aws.bluage.l3.workshop.cbact04c.business.context;
2
3  import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
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

```java
@Component("aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext")
@Import({
    aws.bluage.l3.workshop.cbact04c.business.model.TcatbalFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.XrefFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.AccountFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.TransactFile.class
})
@Lazy
@Scope("prototype")
public class Cbact04cContext extends JicsRuntimeContext {

    @Autowired
    private TcatbalFile tcatbalFile;

    @Autowired
    private XrefFile xrefFile;

    @Autowired
    private DiscgrpFile discgrpFile;

    @Autowired
    private AccountFile accountFile;

    @Autowired
    private TransactFile transactFile;

    private IndexedFile tcatbalFileFile;

    private IndexedFile xrefFileFile;

    private IndexedFile discgrpFileFile;

    private IndexedFile accountFileFile;

    private SequentialFile transactFileFile;

    private TranCatBalRecord tranCatBalRecord;
    private TcatbalfStatus tcatbalfStatus;
    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. Cosa sono i semplificatori di dati in AWS Transform for mainframe

DiscgrpFile.java

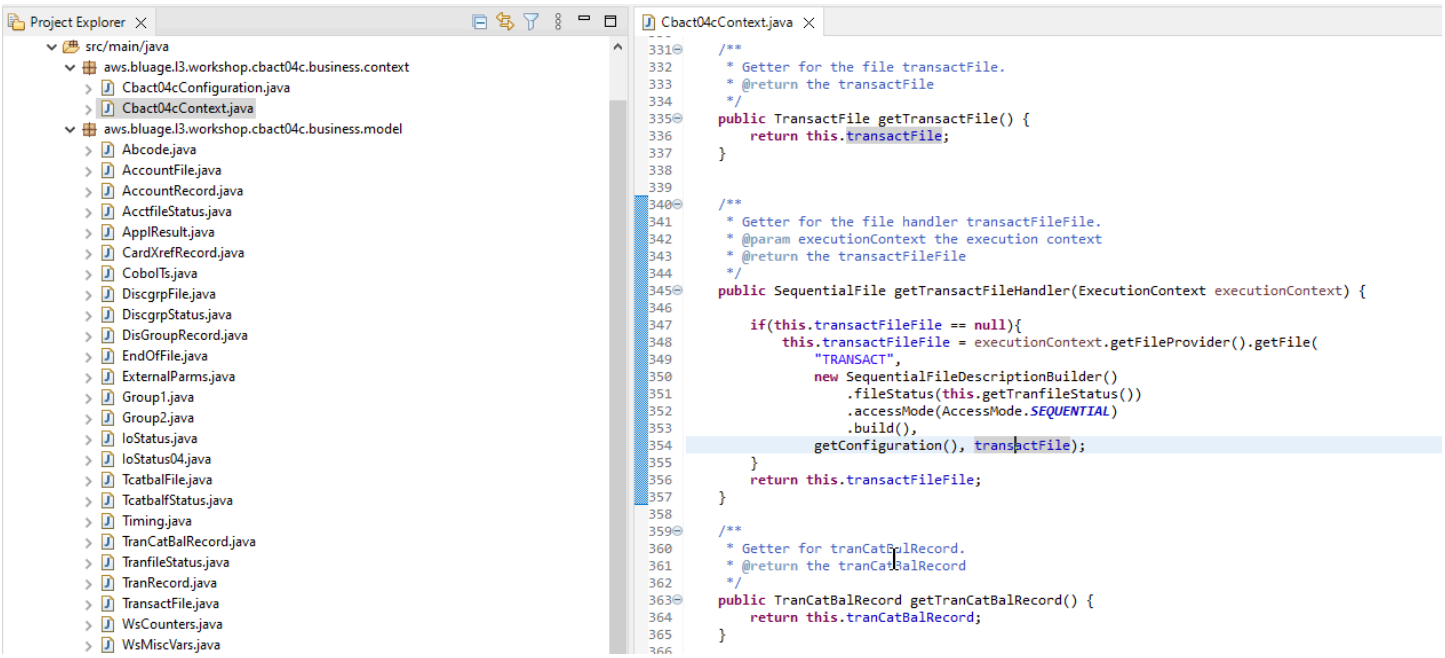
```java
package aws.bluage.l3.workshop.cbact04c.business.model;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary;
import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group;
import com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference;
import com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference;
import com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity;
import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType;
import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Lazy;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

/**
 * Data simplifier file DiscgrpFile.
 *
 * <p>About 'fdDiscgrpRec' field, <br>uml entity: aws.bluage.l3.workshop.cbact04c.business.model.FdDiscgrpRec
 * <br></p>
 *
 */
@Component("aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile")
@Lazy
@Scope("prototype")
public class DiscgrpFile extends RecordEntity {

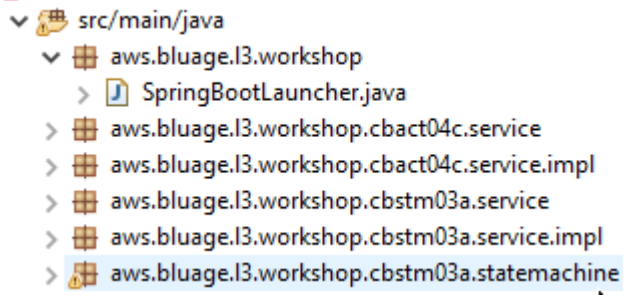
    private final Group root = new Group(getData());
    private final Group fdDiscgrpRec = new Group(root);
    private final Group fdDiscgrpKey = new Group(fdDiscgrpRec);
    private final Elementary fdDisAcctGroupId = new Elementary(fdDiscgrpKey,new AlphanumericType(10));
    private final Elementary fdDisTranTypeCd = new Elementary(fdDiscgrpKey,new AlphanumericType(2));
    private final Elementary fdDisTranCatCd = new Elementary(fdDiscgrpKey,new ZonedType(4, 0, false));
    private final Elementary fdDiscgrpData = new Elementary(fdDiscgrpRec,new AlphanumericType(34));
```

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 un`File`. 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).



Contenuti 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:



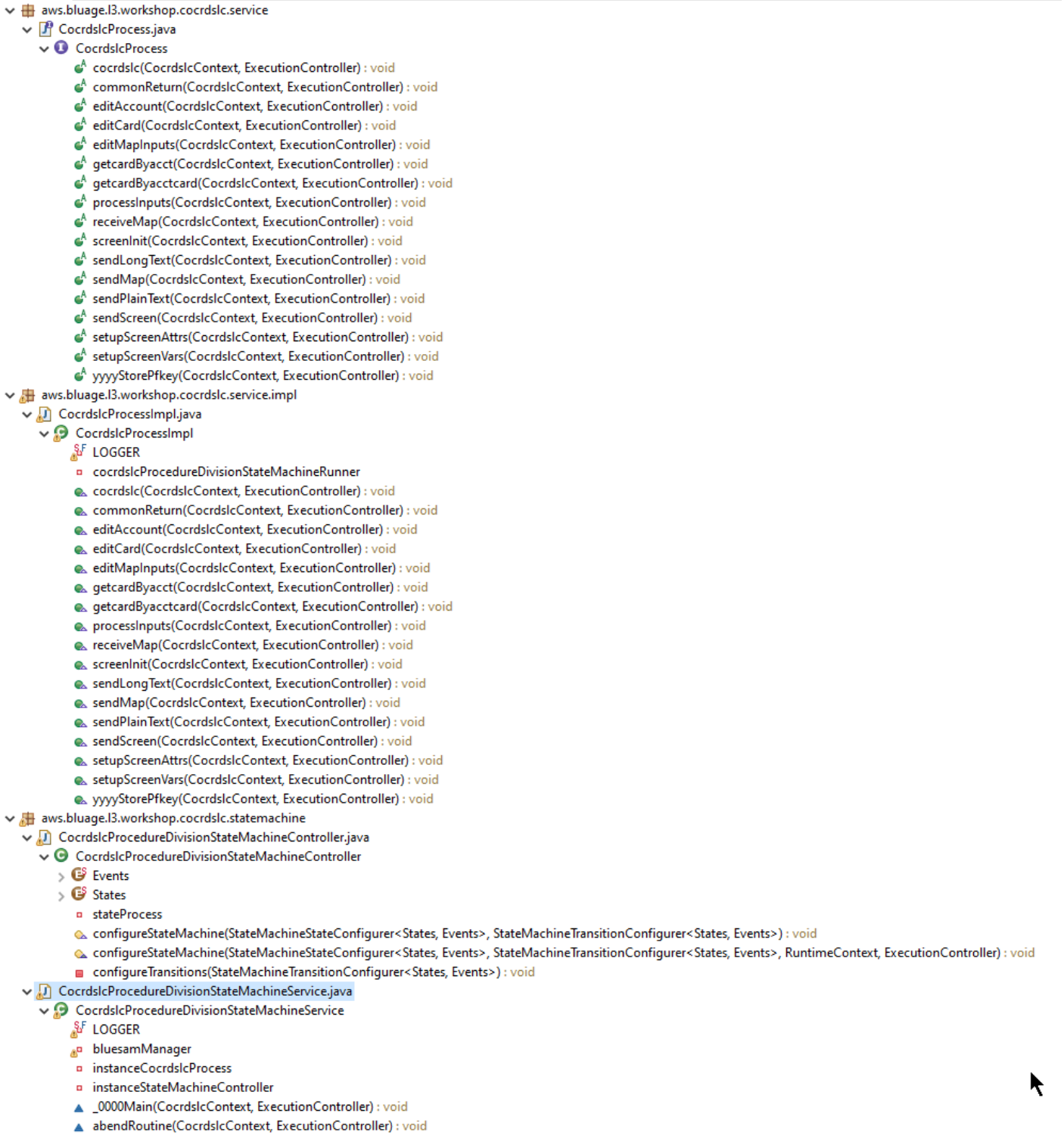
Questa classe è in particolare responsabile di:

- 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:



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 Transform for mainframe:

CocrdslcProcessImpl.java

```java
    /**
     * Process operation sendScreen.
     *
     * @param ctx
     * @param ctrl
     */
    @Override
    public void sendScreen(final CocrdslcContext ctx, final ExecutionController ctrl) {
        ctx.getCcWorkAreas().setCcardNextMapset(ctx.getWsLiterals().getLitThismapset());
        ctx.getCcWorkAreas().setCcardNextMap(ctx.getWsLiterals().getLitThismap());
        ctx.getCarddemoCommarea().setCdemoPgmReenter(true);
        SendMapBuilder.newInstance(ctx.getDfheiblk(), ctx)
        .withMap(ctx.getCcWorkAreas().getCcardNextMap())
        .withMapset(ctx.getCcWorkAreas().getCcardNextMapset())
        .withData(ctx.getGroup1().getCcrdslaoReference())
        .withCursor()
        .withErase()
        .withFreeKB()
        .execute();
        ctx.getWsMiscStorage().setWsRespCd(ctx.getDfheiblk().getEibresp());
    }

    /**
     * Process operation processInputs.
     *
     * @param ctx
     * @param ctrl
     */
    @Override
    public void processInputs(final CocrdslcContext ctx, final ExecutionController ctrl) {
        receiveMap(ctx, ctrl);
        editMapInputs(ctx, ctrl);
        ctx.getCcWorkAreas().setCcardErrorMsg(ctx.getWsMiscStorage().getWsReturnMsg());
        ctx.getCcWorkAreas().setCcardNextProg(ctx.getWsLiterals().getLitThispgm());
        ctx.getCcWorkAreas().setCcardNextMapset(ctx.getWsLiterals().getLitThismapset());
        ctx.getCcWorkAreas().setCcardNextMap(ctx.getWsLiterals().getLitThismap());
    }

```

- *`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 StateMachine framework Spring) per coprire adeguatamente il flusso di esecuzione precedente.

In tal caso, il sottopacchetto statemachine contiene due classi:

- *Program*`ProcedureDivisionStateMachineController`: 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.

CocrdslcProcedureDivisionStateMachineController.java

```
package aws.bluage.l3.workshop.cocrdslc.statemachine;

import aws.bluage.l3.workshop.cocrdslc.business.context.CocrdslcContext;

/**
 * Controller managing the state machine "CocrdslcProcedureDivisionStateMachine" execution.
 */
@Component("aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineController")
@Import({
aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineService.class
})
@Lazy
public class CocrdslcProcedureDivisionStateMachineController extends SimpleStateMachineController<States, Events> {

    /**
     * State machine states.
     */
    public enum States {
        _0000_MAIN_1, _0000_MAIN, ABEND_ROUTINE, FINAL, LOCAL_FINAL
    }

    /**
     * State machine events.
     */
    public enum Events {
        TO__0000_MAIN_1, TO__0000_MAIN, TO_ABEND_ROUTINE, TO_FINAL, TO_LOCAL_FINAL
    }

    /**
     * State machine state process service provider.
     */
    @Autowired
    @Lazy
    private CocrdslcProcedureDivisionStateMachineService stateProcess;

    @Override
    protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
        throw new UnsupportedOperationException("Please use the four arguments configureStateMachine method instead: configureStateMachine(StateMachineStateConfigurer<States, Events> states, "
                + "StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl)");
    }

    @Override
    protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl) throws Exception {
        StateConfigurer<States, Events> configurer = states.withStates();
        configurer.initial(States._0000_MAIN_1).end(States.FINAL);
        configurer.state(States._0000_MAIN_1);
        configurer.state(States.FINAL);

        StateConfigurer<States, Events> subConfigurer = states.withStates().parent(States._0000_MAIN_1);
        subConfigurer.initial(States._0000_MAIN).end(States.LOCAL_FINAL);
        CocrdslcContext lctx = (CocrdslcContext) ctx;
        subConfigurer.state(States._0000_MAIN, buildAction(() -> {stateProcess._0000Main(lctx, ctrl);}), null);
        subConfigurer.state(States.ABEND_ROUTINE, buildAction(() -> {stateProcess.abendRoutine(lctx, ctrl);}), null);

        configureTransitions(transitions);
    }

    /**
     * Declare state machine transitions.
     * @param transitions the transitions configuration helper
     */
    private void configureTransitions(StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
        transitions.withLocal().source(States._0000_MAIN_1).target(States.ABEND_ROUTINE).event(Events.TO_ABEND_ROUTINE);
        transitions.withExternal().source(States.ABEND_ROUTINE).target(States.FINAL).event(Events.TO_FINAL);
    }
}

```

Il controller della macchina a stati definisce i possibili diversi stati e le transizioni tra di essi, che riproducono 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);
```

- *Program*ProcedureDivisionStateMachineService: 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:     COCRDSLC.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

```
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
```

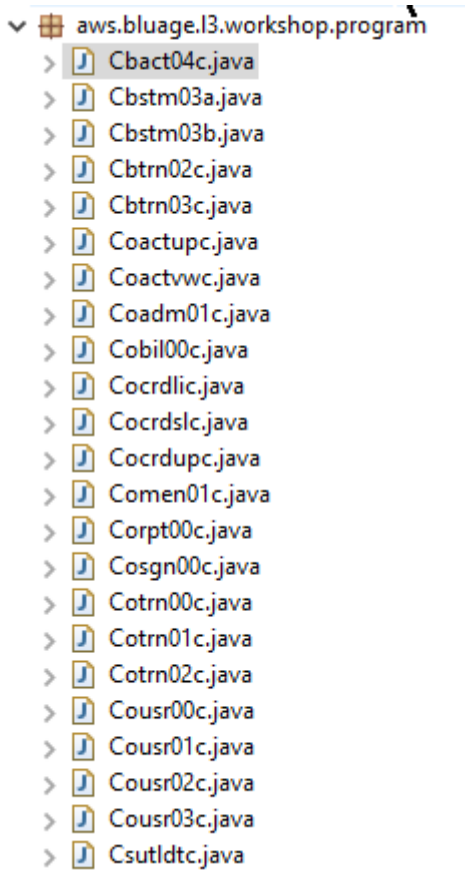
Il servizio statemachine effettua anche chiamate all'implementazione del servizio di processo descritta in precedenza:

CocrdslcProcedureDivisionStateMachineService.java

```
166
167         /*
168         ************************************************************
169         COMING FROM CREDIT CARD LIST SCREEN
170         SELECTION CRITERIA ALREADY VALIDATED
171         ************************************************************ */
172     } else if (ctx.getCarddemoCommarea().isCdemoPgmEnter() && DataUtils.compare(ctx.getCarddemoCommarea().getCdemoFromProgramReference(), ctx.getWsLiterals().getLitCclistpgmReference()) == 0) {
173         ctx.getWsMiscStorage().setInputOk(true);
174         ctx.getCcWorkAreas().setCcAcctIdN(ctx.getCarddemoCommarea().getCdemoAcctId());
175         ctx.getCcWorkAreas().setCcCardNumN(ctx.getCarddemoCommarea().getCdemoCardNum());
176         instanceCocrdslcProcess.getcardByacctcard(ctx, ctrl);
177         instanceCocrdslcProcess.sendMap(ctx, ctrl);
178         instanceCocrdslcProcess.commonReturn(ctx, ctrl);
179     } else if (ctx.getCarddemoCommarea().isCdemoPgmEnter()) {
180
181         /*
182         ************************************************************
183         COMING FROM SOME OTHER CONTEXT
184         SELECTION CRITERIA TO BE GATHERED
185         ************************************************************ */
186         instanceCocrdslcProcess.sendMap(ctx, ctrl);
187         instanceCocrdslcProcess.commonReturn(ctx, ctrl);
188     } else if (ctx.getCarddemoCommarea().isCdemoPgmReenter()) {
189         instanceCocrdslcProcess.processInputs(ctx, ctrl);
190         if (ctx.getWsMiscStorage().isInputError()) {
191             instanceCocrdslcProcess.sendMap(ctx, ctrl);
192             instanceCocrdslcProcess.commonReturn(ctx, ctrl);
193         } else {
194             instanceCocrdslcProcess.getcardByacctcard(ctx, ctrl);
195             instanceCocrdslcProcess.sendMap(ctx, ctrl);
196             instanceCocrdslcProcess.commonReturn(ctx, ctrl);
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.getCcWorkAreas().setCcardErrorMsg(ctx.getWsMiscStorage().getWsReturnMsg());
211         instanceCocrdslcProcess.sendMap(ctx, ctrl);
212         instanceCocrdslcProcess.commonReturn(ctx, ctrl);
213     }
214     instanceCocrdslcProcess.commonReturn(ctx, ctrl);
```

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

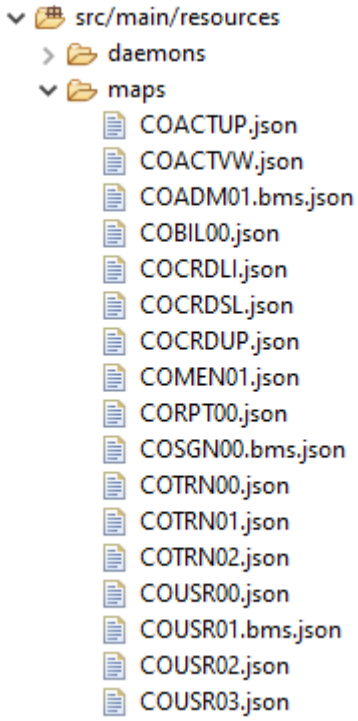
dettagli su questo argomento più avanti). Ogni classe implementa l'`Program`interfaccia, indicatore di un punto di ingresso del programma.



Altri manufatti

- Compagni BMS MAPs

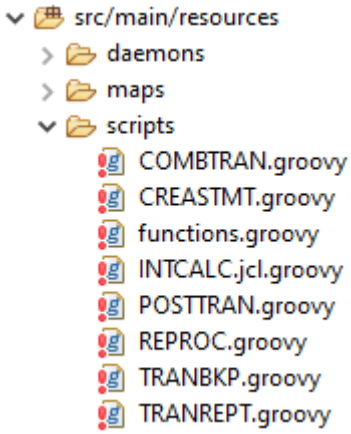
  Oltre agli elementi relativi al programma, il progetto di servizio può contenere altri artefatti per vari scopi. Nel caso della modernizzazione di un'applicazione online CICS, il processo di modernizzazione produce un file json e inserisce nella cartella map della cartella/: src/main/resources



Il AWS Transform for mainframe runtime utilizza questi file json per associare i record utilizzati dall'istruzione SEND MAP ai campi dello schermo.

- Script Groovy

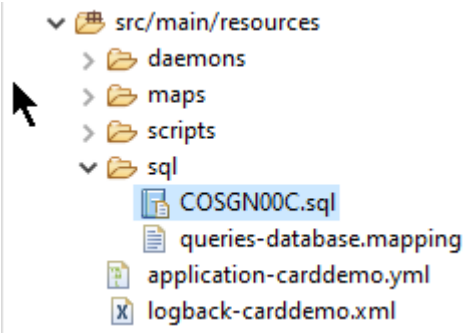
  Se l'applicazione precedente aveva script JCL, questi sono stati modernizzati come script groovy, archiviati nella src/main/resources/scripts cartella/(maggiori informazioni su quella posizione specifica più avanti):



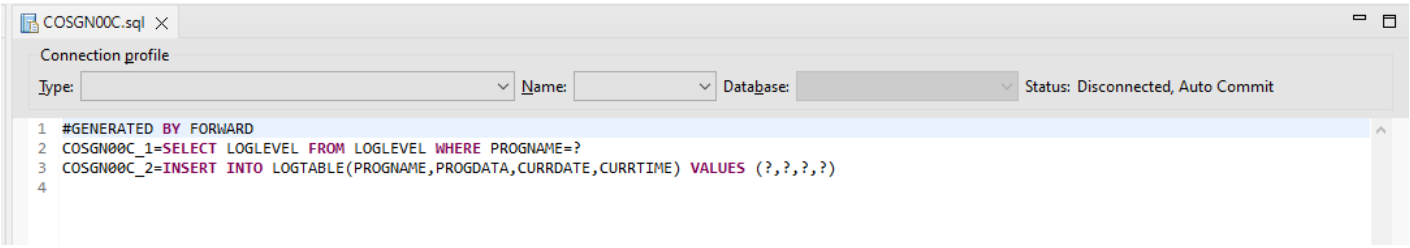
  Questi script vengono utilizzati per avviare processi in batch (carichi di lavoro di elaborazione dati dedicati, non interattivi e che richiedono un uso intensivo della CPU).

- File SQL

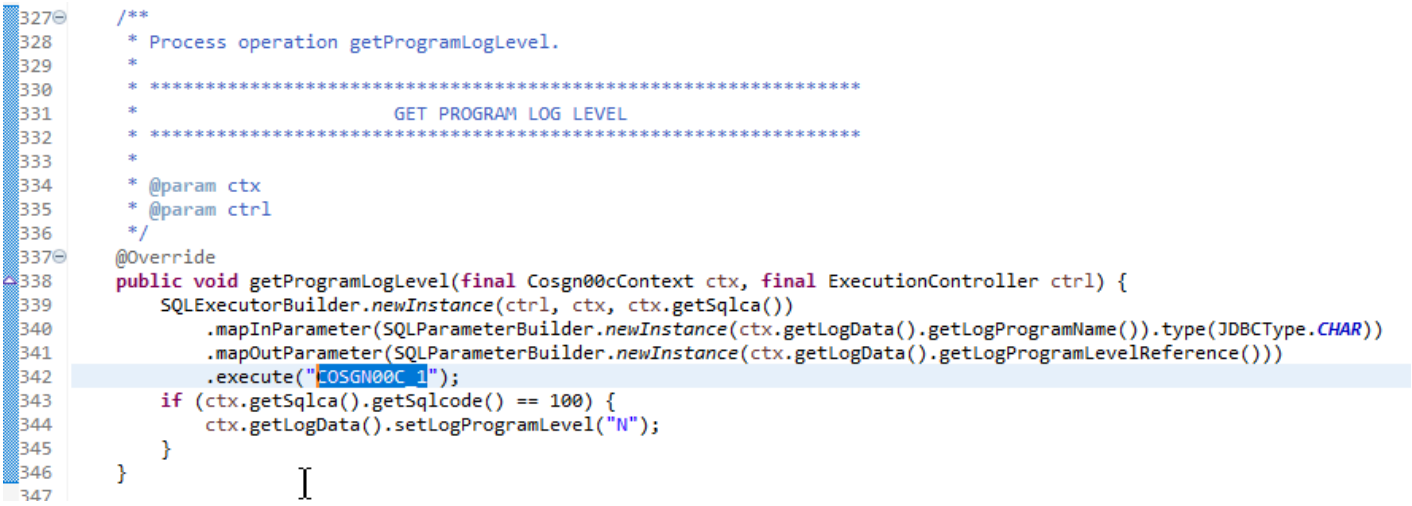
Se l'applicazione precedente utilizzava query SQL, le corrispondenti query SQL modernizzate sono state raccolte in file di proprietà dedicati, con lo schema di denominazione program .sql, dove program è il nome del programma che utilizza tali query.



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:

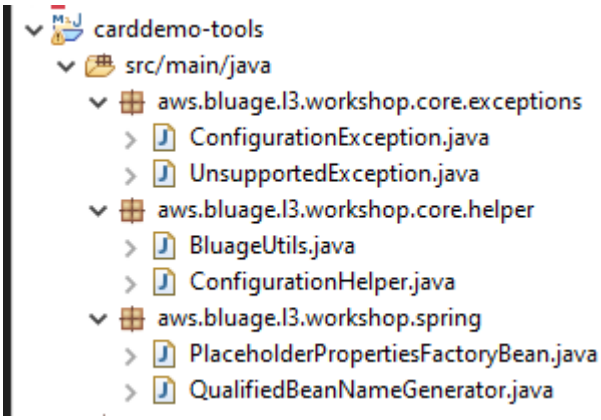


Ad esempio, il programma COSGN00 C sta eseguendo la query con la chiave "COSGN00C_1" (la prima voce nel file sql):



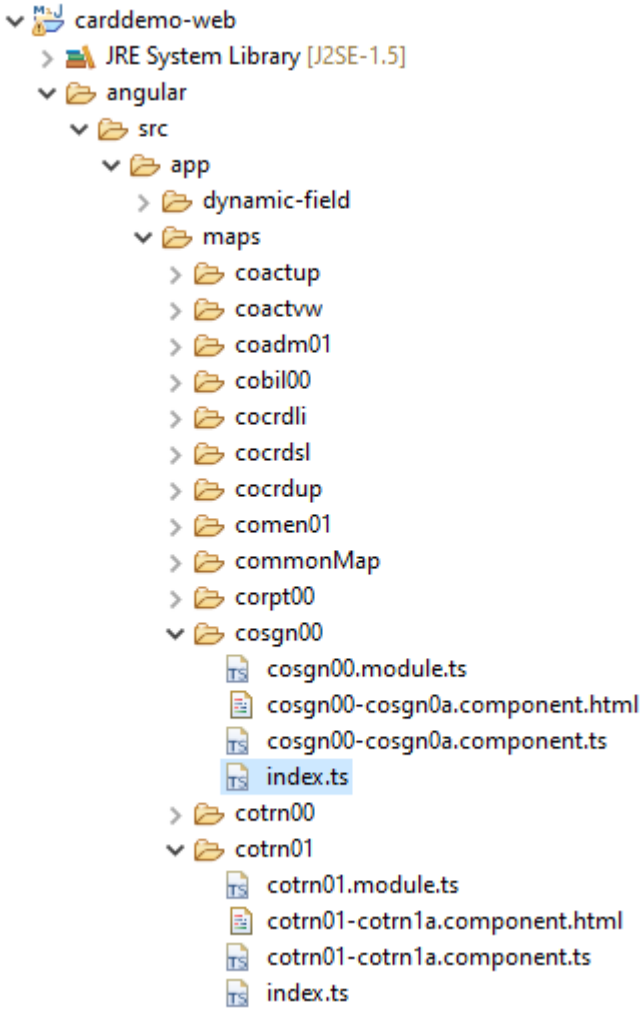
Utilità, contenuti del progetto

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' COBOL/CICS applicazione in esecuzione su un mainframe. Il sistema CICS utilizza per rappresentare le schermate dell'interfaccia MAPs utente. Gli elementi moderni corrispondenti saranno, per ogni mappa, un file html accompagnato da file Typescript:



Il progetto web si occupa solo dell'aspetto front-end dell'applicazione. Il progetto di servizio, che si basa sui progetti di utilità ed entità, fornisce i servizi di backend. Il collegamento tra il front-end e il backend viene effettuato tramite l'applicazione web denominata Gapwalk-Application, che fa parte dello standard Transform for mainframe runtime distribution. 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'`Program`interfaccia, 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 con le voci del programma, ogni programma viene registrato utilizzando i propri 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

```java
package aws.bluage.l3.workshop.program;

import aws.bluage.l3.workshop.SpringBootLauncher;

/**
 * Reference the spring application of program CBACT04C.
 *  Provides an access to the contained program for the run unit.
 */
@Component
@Import({
    aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cConfiguration.class,
    aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext.class,
    aws.bluage.l3.workshop.cbact04c.service.impl.Cbact04cProcessImpl.class
})
public class Cbact04c implements Program {
    /**
     * Unique identifiers for the contained program.
     */
    private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("CBACT04C").collect(Collectors.toSet()));

    /**
     * Main program identifier for the contained program.
     */
    private static final String programIdentifier = "CBACT04C";
    @Autowired
    PlatformTransactionManager transactionManager;

    @Autowired
    Map<String, DataSource> datasources;
    @Autowired
    BeanFactory beanFactory;
    /**
     * {@inheritDoc}
     */
    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        return SpringBootLauncher.getCac();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void updateExecutionContext(ExecutionContext executionContext) {
        executionContext.setDatasources(datasources);
        executionContext.setDatabaseSupport(ExecutionContext.DatabaseSupport.POSTGRE);
        executionContext.setSqlcaVersion(ExecutionContext.SqlcaVersion.getEnum("ansi-comp5"));
        executionContext.setTransactionManager(transactionManager);
        executionContext.setUseSQLDateNewParadigm(true);
        executionContext.setUseSQLTrimStringType(false);
    }

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

In questo esempio, il programma viene registrato una sola volta, con il nome 'CBACT04C' (guarda 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 utilità AWS Transform for mainframe web applications, che fanno parte della distribuzione di runtime AWS Transform for mainframe. Ad esempio, la Gapwalk-Utility-Pgm webapp fornisce gli equivalenti

funzionali delle utilità di z/OS sistema (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 log 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 (eccetto lo script riservato special functions.groovy) vengono registrati in`ScriptRegistry`, utilizzando il loro nome breve (la parte del nome del file di script che precede il primo carattere del 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: tipicamente /.groovy. scripts/A/myscript.groovy and /scripts/B/myscript.groovy will lead to / scripts/B/myscript.groovy overwriting /scripts/A/myscript

Gli script groovy nella cartella/src/main/resources/daemonsvengono 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 CBST03 A chiama il programma CBST03 B come sottoprogramma, passandogli dei parametri:

CbstmO3aProcessImpl.java

```
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      }
```

- 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 chiamante.
- 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 B.CBL corrispondente: CBSTM03

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 'O'.
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 CBSTM03 B ne 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 a`Record`, 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 a`Record`, 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

```groovy
// Import
import com.netfective.bluage.gapwalk.rt.provider.ScriptRegistry
import com.netfective.bluage.gapwalk.rt.call.MainProgramRunner
import com.netfective.bluage.gapwalk.io.support.FileConfigurationUtils
import com.netfective.bluage.gapwalk.rt.job.support.DefaultJobContext
import com.netfective.bluage.gapwalk.rt.utils.GroovyUtils
import com.netfective.bluage.gapwalk.rt.io.support.FileConfiguration
import com.netfective.bluage.gapwalk.rt.shared.AbendException
import com.netfective.bluage.gapwalk.rt.call.exception.GroovyExecutionException
// Variables
mpr = applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController", MainProgramRunner.class)
TreeMap mapTransfo = [:]
```

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

REPROC.groovy

```
//*******************************************************************
//*                              STEPS                              *
//*******************************************************************
// STEP PRC001 - PGM - IDCAMS*****************************************
def stepPRC001(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("PRC001", "IDCAMS", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .systemOut("SYSPRINT")
                        .output("*")
                        .build()
                        .bluesam("FILEIN")
                        .dataset("NULLFILE")
                        .disposition("SHR")
                        .build()
                        .bluesam("FILEOUT")
                        .dataset("NULLFILE")
                        .disposition("SHR")
                        .build()
                        .fileSystem("SYSIN")
                        .path("&CNTLLIB(REPROCT)")
                        .disposition("SHR")
                        .build()
                        .getFileConfigurations(fcmap))
                    .withParameters(params)
                    .runProgram("IDCAMS")
                })
        }
    }
}
```

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.

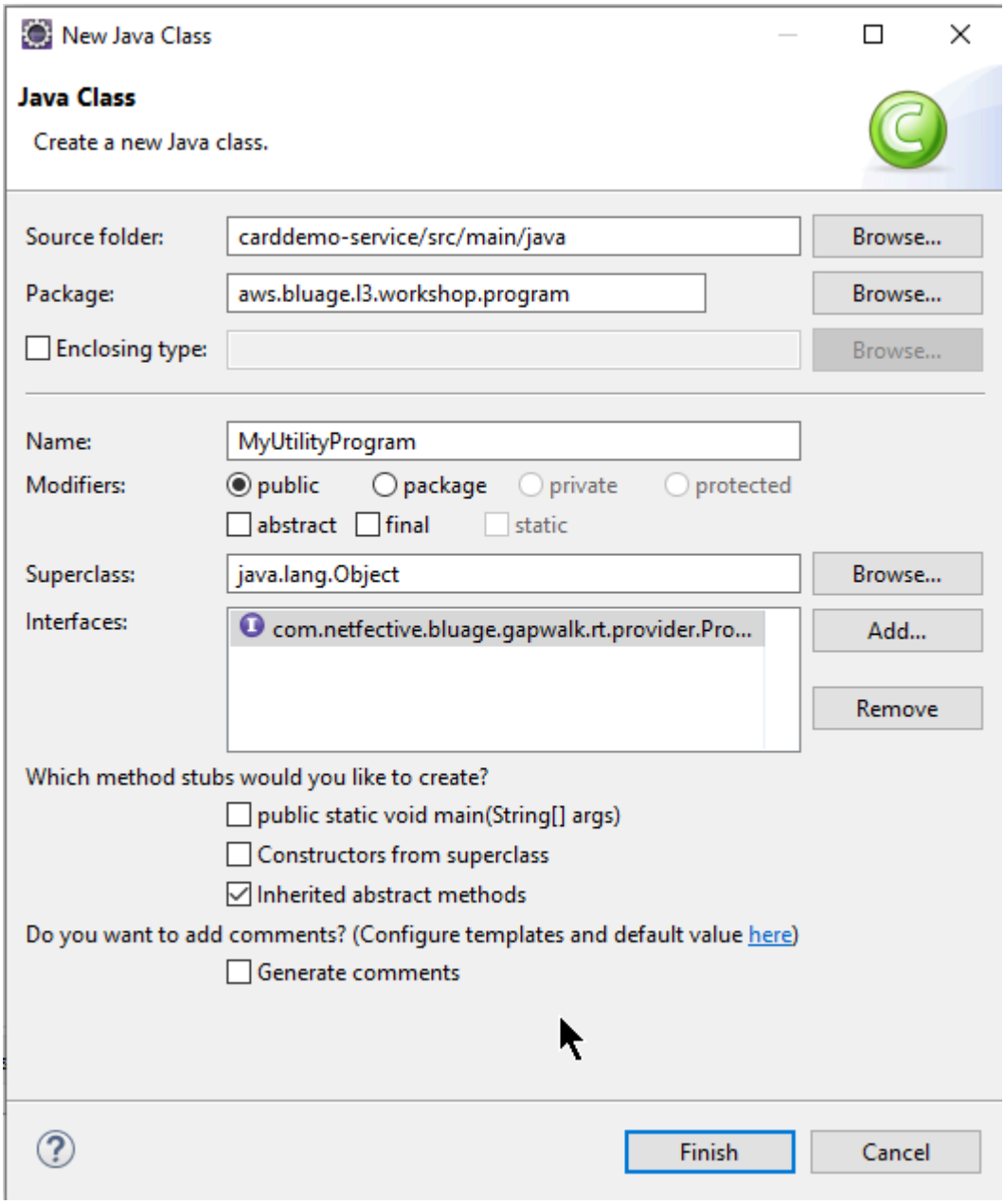
In tal caso, potresti dover scrivere il programma mancante, in java, da solo (supponendo che tu abbia una conoscenza sufficiente di quale dovrebbe essere il comportamento previsto del programma, del layout di memoria degli eventuali argomenti del programma e così via). Il programma java deve essere conforme 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'`Program`interfaccia 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'`Program`interfaccia:



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

MyUtilityProgram.java

```java
package aws.bluage.l3.workshop.program;

import java.util.Set;

public class MyUtilityProgram implements Program {

    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public Set<String> getProgramIdentifiers() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public Context getContext() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void run(ExecutionController ctrl) {
        // TODO Auto-generated method stub

    }

}
```

Integrazione primaverile

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

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

import com.netfective.bluage.gapwalk.rt.call.ExecutionController;
import com.netfective.bluage.gapwalk.rt.context.Context;
import com.netfective.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 coprire un programma esistente sostitutivo di un programma legacy; 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() {

    @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 Transform for mainframe 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.

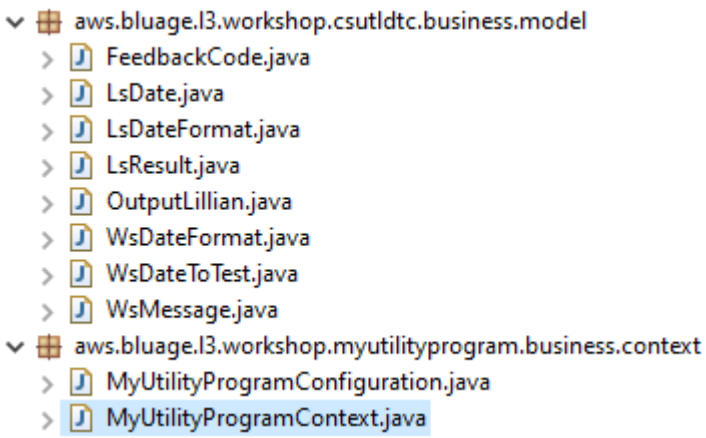
Facendo riferimento aClassi relative al programma, puoi vedere che un programma richiede almeno due classi complementari:

- 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:



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

```java
package aws.bluage.l3.workshop.myutilityprogram.business.context;

import java.nio.charset.Charset;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Lazy;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.datasimplifier.configuration.ConfigurationBuilder;

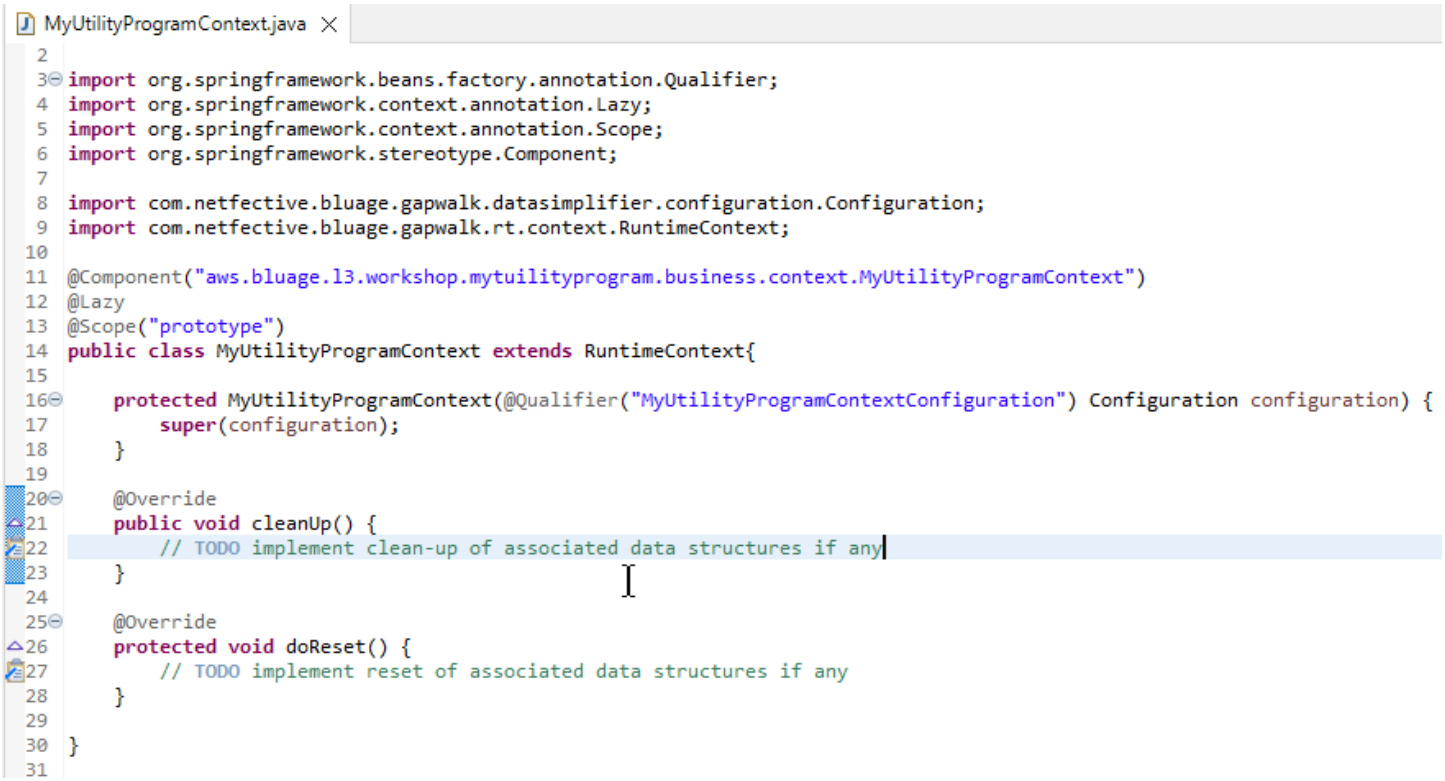
/**
 * Creates Datasimplifier configuration for the MyUtilityProgram context.
 */
@org.springframework.context.annotation.Configuration
@Lazy
public class MyUtilityProgramConfiguration {

    @Bean(name = "MyUtilityProgramContextConfiguration")
    public Configuration configuration() {
        return new ConfigurationBuilder()
                .encoding(Charset.forName("CP1047"))
                .humanReadableEncoding(Charset.forName("ISO-8859-15"))
                .initDefaultByte(0)
                .build();
    }
}
```

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 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:



```java
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Lazy;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.rt.context.RuntimeContext;

@Component("aws.bluage.l3.workshop.mytuilityprogram.business.context.MyUtilityProgramContext")
@Lazy
@Scope("prototype")
public class MyUtilityProgramContext extends RuntimeContext{

    protected MyUtilityProgramContext(@Qualifier("MyUtilityProgramContextConfiguration") Configuration configuration) {
        super(configuration);
    }

    @Override
    public void cleanUp() {
        // TODO implement clean-up of associated data structures if any
    }

    @Override
    protected void doReset() {
        // TODO implement reset of associated data structures if any
    }

}
```

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

```java
import aws.bluage.l3.workshop.SpringBootLauncher;

@Component
@Import({
    aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramConfiguration.class,
    aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class
})
public class MyUtilityProgram implements Program {

    @Autowired
    BeanFactory beanFactory;

    /**
     * Unique identifiers for the contained program.
     */
    private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("MYUTILPG").collect(Collectors.toSet()));

    private static final String programIdentifier = "MYUTILPG";

    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        return SpringBootLauncher.getCac();
    }

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

    /**
     * {@inheritDoc}
     */
    @Override
    public String getProgramMainIdentifier() {
        return programIdentifier;
    }


    @Override
    public Context getContext() {
        return ProgramContextStore.getOrCreate(
        getProgramMainIdentifier(),
        aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class,
        beanFactory);
    }

```

Note:

- Il metodo getContext deve essere implementato rigorosamente come mostrato, utilizzando una delega al getOrCreate metodo della `ProgramContextStore` classe e al cablato automatico Spring. `BeanFactory` Un singolo identificatore di programma viene utilizzato per memorizzare il contesto del programma in`ProgramContextStore`; questo identificatore è 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 (vediContenuti 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 Transform for mainframe e di terze parti da utilizzare nelle applicazioni modernizzate.

AWS Transform for mainframe: mappature dei nomi completamente qualificate

Nome breve	Nome completamente qualificato
`CallBuilder`	`com.netfective.bluage.gapwalk.runtime.statements.CallBuilder`
`Configuration`	`com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration`

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.ExecutionControllerImpl
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.ProgramContextStore
ProgramRegistry	com.netfective.bluage.gapwalk.rt.provider.ProgramRegistry
Record	com.netfective.bluage.gapwalk.datasimplifier.data.Record
RecordEntity	com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity
RuntimeContext	com.netfective.bluage.gapwalk.rt.context.RuntimeContext

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

Cosa sono i semplificatori di dati in AWS Transform for mainframe

Nei sistemi mainframe e midrange (denominati nel seguente argomento 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 insiemi di byte contigui (LINKAGE) o i read/write dati dei file nello stesso modo (record), interpretando tali intervalli di memoria con campi digitati 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 Transform for mainframe 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 sia PL1 per la modernizzazione del layout dei dati 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 Transform for mainframe. 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 Transform for mainframe a partire dai costrutti legacy corrispondenti. È comunque utile conoscerne la struttura principale e l'API, in modo da capire come vengono utilizzati 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 Transform for mainframe 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` (o`Filler`, 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 in un array di byte). from/to

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

Tipo COBOL	Tipo di semplificatore di dati
`PIC X(n)`	`AlphanumericType`
`PIC 9(n)`	`ZonedType`
`PIC 9(n) COMP-3`	`PackedType`

Tipo COBOL	Tipo di semplificatore di dati
`PIC 9(n) COMP-5`	`BinaryType`

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.

`Group`i 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 + (size in bytes of first field)` offset, ecc. Vengono utilizzati per rappresentare sequenze di campi COBOL nello stesso campo contenitore.

`Union`i 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 Transform for mainframe 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. Allo stesso modo, AWS Transform per mainframe APIs e il codice che manipola la memoria di basso livello (come

argomenti dei programmi, I/O record di file, area di comunicazione CICS, memoria allocata...) spesso si aspettano un come 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 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`

Si noti 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 Transform for mainframe (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 sottoclassi e `RecordEntity` 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`	`com.netfective.bluage.gapwalk.datasimplifier.elementary.Alphanumeric`
`AlphanumericType`	`com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType`
`BinaryType`	`com.netfective.bluage.gapwalk.datasimplifier.metadata.type.BinaryType`
`Bindable`	`com.netfective.bluage.gapwalk.datasimplifier.data.Bindable`
`Elementary`	`com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary`
`ElementaryRangeReference`	`com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference`
`Filler`	`com.netfective.bluage.gapwalk.datasimplifier.data.structure.Filler`
`Group`	`com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group`

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

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

AWS Transform per mainframe Blusam

Nei sistemi mainframe (denominati «legacy» nel seguente argomento), i dati aziendali vengono spesso archiviati utilizzando VSAM (Virtual Storage Access Method). I dati vengono archiviati in «record» (matrici di byte), appartenenti a un «set di dati».

Esistono quattro organizzazioni di set di dati:

- KSDS: set di dati con sequenza di tasti: i record sono indicizzati mediante una chiave primaria (non sono consentite chiavi duplicate) e, facoltativamente, chiavi «alternative» aggiuntive. Tutti i valori delle chiavi sono sottoinsiemi dell'array di record byte, ogni chiave è definita da:
  - un offset (in base a 0, 0 è l'inizio del contenuto dell'array di record byte, misurato in byte)
  - una lunghezza (espressa in byte)
  - se tollera o meno valori duplicati
- ESDS: set di dati in sequenza di ingresso: l'accesso ai record avviene per lo più in sequenza (in base all'ordine di inserimento nel set di dati) ma è possibile accedervi utilizzando chiavi alternative aggiuntive;
- RRDS: set di dati Relative Records: i record sono accessibili tramite «salti», utilizzando numeri di record relativi; i salti possono essere eseguiti in avanti o indietro;
- LDS: set di dati lineari: nessun record, semplicemente un flusso di byte, organizzato in pagine. Utilizzato principalmente per scopi interni su piattaforme legacy.

Quando si modernizzano le applicazioni legacy, utilizzando l'approccio AWS Transform for mainframe refactoring, le applicazioni modernizzate non sono più destinate ad accedere ai dati archiviati in VSAM, preservando al contempo la logica di accesso ai dati. Il Blusam componente è la risposta:

consente di importare dati dalle esportazioni di set di dati VSAM legacy, fornisce un'API per l'applicazione modernizzata per manipolarli insieme a un'applicazione web di amministrazione dedicata. Per informazioni, consulta the section called "BlusamConsole di amministrazione".

> **Note**
>
> Blusamsupporta solo KSDS, ESDS e RRDS.

L'BlusamAPI consente di preservare la logica di accesso ai dati (letture sequenziali, casuali e relative; inserimento, aggiornamento ed eliminazione dei record), mentre l'architettura dei componenti, basata su una combinazione di strategie di caching e archiviazione basata su RDBMS, consente operazioni ad alto rendimento con risorse limitate. I/O

infrastruttura Blusam

Blusamsi affida a PostgreSQL RDBMS per l'archiviazione dei set di dati, sia per i dati dei record non elaborati che per gli indici delle chiavi (se applicabile). L'opzione preferita è utilizzare il motore compatibile con Amazon Aurora PostgreSQL. Gli esempi e le illustrazioni in questo argomento si basano su questo motore.

> **Note**
>
> All'avvio del server, il Blusam runtime verifica la presenza di alcune tabelle tecniche obbligatorie e le crea se non possono essere trovate. Di conseguenza, al ruolo utilizzato nella configurazione per accedere al Blusam database devono essere concessi i diritti per creare, aggiornare ed eliminare le tabelle del database (sia le righe che le definizioni delle tabelle stesse). Per informazioni su come disattivareBlusam, vederethe section called "Configurazione di Blusam".

Caching

Oltre allo storage stesso, Blusam funziona più velocemente se abbinato a un'implementazione della cache.

Attualmente sono supportati due motori di cache EhCache e Redis, ognuno con il proprio caso d'uso:

- EhCache : cache locale volatile incorporata autonoma

  - NON idoneo per l'implementazione di AWS ambienti gestiti per la modernizzazione del mainframe.
  - In genere viene utilizzato quando un singolo nodo, ad esempio un singolo server Apache Tomcat, viene utilizzato per eseguire le applicazioni modernizzate. Ad esempio, il nodo potrebbe essere dedicato all'hosting di attività di lavoro in batch.
  - Volatile: l'istanza EhCache della cache è volatile; il suo contenuto andrà perso alla chiusura del server.
  - Incorporato: il server EhCache e il server condividono lo stesso spazio di memoria JVM (da tenere in considerazione quando si definiscono le specifiche per la macchina di hosting).
- Redis: cache persistente condivisa
  - Idoneo per l'implementazione di ambienti gestiti per la modernizzazione del AWS mainframe.
  - Utilizzato in genere in situazioni con più nodi, in particolare quando diversi server sono protetti da un sistema di bilanciamento del carico. Il contenuto della cache è condiviso tra tutti i nodi.
  - Il Redis è persistente e non è legato ai cicli di vita dei nodi. È in esecuzione su un computer o servizio dedicato (ad esempio, Amazon ElastiCache). La cache è remota per tutti i nodi.

Blocco

Per gestire l'accesso simultaneo a set di dati e record, Blusam si affida a un sistema di chiusura configurabile. Il blocco può essere applicato a entrambi i livelli: set di dati e record:

- Il blocco di un set di dati a scopo di scrittura impedirà a tutti gli altri client di eseguire operazioni di scrittura su di esso, a qualsiasi livello (set di dati o record).
- Il blocco a livello di record per la scrittura impedirà ad altri client di eseguire operazioni di scrittura solo sul record specificato.

La configurazione del sistema di Blusam blocco deve essere eseguita in base alla configurazione della cache:

- Se EhCache viene scelta come implementazione della cache, non è necessaria alcuna ulteriore configurazione di blocco in quanto deve essere utilizzato il sistema di blocco in memoria predefinito.
- Se si sceglie Redis come implementazione della cache, è necessaria una configurazione di blocco basata su Redis, per consentire l'accesso simultaneo da più nodi. La cache Redis utilizzata per i

blocchi non deve essere la stessa utilizzata per i set di dati. Per informazioni sulla configurazione di un sistema di blocco basato su Redis, consulta. the section called "Configurazione di Blusam"

Blusamelementi intrinseci e migrazione dei dati da sistemi preesistenti

Archiviazione di set di dati: record e indici

Ogni set di dati legacy, una volta importato inBlusam, verrà archiviato in una tabella dedicata; ogni riga della tabella rappresenta un record, utilizzando due colonne:

- La colonna ID numerica, di tipo big integer, è la chiave primaria della tabella e viene utilizzata per memorizzare il Relative Byte Address (RBA) del record. L'RBA rappresenta l'offset in byte dall'inizio del set di dati e inizia da 0.
- La colonna di record dell'array di byte, utilizzata per memorizzare il contenuto del record non elaborato.

Vedi ad esempio il contenuto di un set di dati KSDS utilizzato nell' CardDemo applicazione:

```
1  SELECT * FROM public.aws_m2_carddemo_acctdata_vsam_ksds
2  ORDER BY id ASC
```

Data output | Messages | Notifications

	id [PK] bigint	record bytea
1	0	[binary data]
2	300	[binary data]
3	600	[binary data]
4	900	[binary data]
5	1200	[binary data]
6	1500	[binary data]
7	1800	[binary data]
8	2100	[binary data]
9	2400	[binary data]
10	2700	[binary data]
11	3000	[binary data]
12	3300	[binary data]
13	3600	[binary data]

- Questo particolare set di dati ha record a lunghezza fissa, la cui lunghezza è di 300 byte (quindi la raccolta di ID è multipli di 300).
- Per impostazione predefinita, lo strumento pgAdmin utilizzato per interrogare i database PostgreSQL non mostra il contenuto delle colonne dell'array di byte, ma stampa invece un'etichetta [dati binari].
- Il contenuto non elaborato del record corrisponde al set di dati non elaborato esportato dalla versione precedente, senza alcuna conversione. In particolare, non avviene alcuna conversione del set di caratteri; ciò implica che le parti alfanumeriche del record dovranno essere decodificate da applicazioni modernizzate che utilizzano il set di caratteri legacy, molto probabilmente una variante EBCDIC.

Per quanto riguarda i metadati dei set di dati e gli indici delle chiavi: ogni set di dati è associato a due righe nella tabella denominata. `metadata` Questa è la convenzione di denominazione predefinita. Per informazioni su come personalizzarla, consultathe section called "Configurazione di Blusam".

	name [PK] text	metadata oid
1	AWS_M2_CARDDEMO_ACCTDATA_VSAM_KSDS	66320
2	AWS_M2_CARDDEMO_ACCTDATA_VSAM_KSDS___internal	66321

- La prima riga ha il nome del set di dati come valore della colonna del nome. La colonna dei metadati è una colonna binaria che contiene una serializzazione binaria dei metadati generali del set di dati specificato. Per informazioni dettagliate, vedi the section called "Attributi generali dei metadati dei set di dati".
- La seconda riga contiene il nome del set di dati con il suffisso `___internal'` come valore della colonna del nome. Il contenuto binario della colonna di metadati dipende dal «peso» del set di dati.
  - Per small/medium i set di dati, il contenuto è una serializzazione compressa di:
    - definizione delle chiavi utilizzate dal set di dati; definizione della chiave primaria (per KSDS) e definizioni di chiavi alternative, se applicabile (per KSDS/ESCS)
    - gli indici chiave, se applicabile (KSDS/ESDS con definizioni di chiavi alternative): utilizzati per la navigazione indicizzata dei record; l'indice delle chiavi associa un valore chiave all'RBA di un record;
    - mappa della lunghezza dei record: utilizzata per la navigazione sequenziale/relativa dei record;
  - Per set Large/Very di dati di grandi dimensioni, il contenuto è una serializzazione compressa di:
    - definizione delle chiavi utilizzate dal set di dati; definizione della chiave primaria (per KSDS) e definizioni di chiavi alternative, se applicabile (per KSDS/ESCS)

Inoltre, gli indici dei set di dati di large/very grandi dimensioni (se applicabile) vengono archiviati utilizzando un meccanismo di impaginazione; le serializzazioni binarie delle pagine indice vengono archiviate come righe di una tabella dedicata (una tabella per chiave del set di dati). Ogni pagina di indici è memorizzata in una riga, con le seguenti colonne:

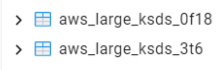
- id: identificatore tecnico della pagina degli indici (chiave numerica primaria);
- firstkey: valore binario del primo valore di chiave (più basso) memorizzato nella pagina degli indici;
- lastkey: valore binario dell'ultimo valore di chiave (più alto) memorizzato nella pagina degli indici;
- metadati: serializzazione binaria compressa della pagina degli indici (mappatura dei valori chiave ai record). RBAs

	id [PK] bigint	firstkey bytea	lastkey bytea	metadata oid
1	1	[binary data]	[binary da...	6458928
2	2	[binary data]	[binary da...	6458929
3	3	[binary data]	[binary da...	6458930
4	4	[binary data]	[binary da...	6458931
5	5	[binary data]	[binary da...	6458932
6	6	[binary data]	[binary da...	6458933
7	7	[binary data]	[binary da...	6458934
8	8	[binary data]	[binary da...	6458935
9	9	[binary data]	[binary da...	6458936

Il nome della tabella è una concatenazione del nome del set di dati e del nome interno della chiave, che contiene informazioni sulla chiave, ad esempio l'offset della chiave, se la chiave accetta i duplicati (impostato su true per consentire i duplicati) e la lunghezza della chiave. Ad esempio, consideriamo un set di dati denominato "AWS_LARGE_KSDS» con le due chiavi definite seguenti:

- chiave primaria [offset: 0, duplicati: false, lunghezza: 18]
- chiave alternativa [offset: 3, duplicati: true, lunghezza: 6]

In questo caso, le tabelle seguenti memorizzano gli indici relativi alle due chiavi.



Ottimizzazione del I/O throughput utilizzando il meccanismo write-behind

Per ottimizzare le prestazioni delle operazioni di inserimento/aggiornamento/eliminazione, il Blusam motore si affida a un meccanismo write-behind configurabile. Il meccanismo si basa su un pool di thread dedicati che gestiscono le operazioni di persistenza utilizzando query di aggiornamento in blocco, per massimizzare la velocità di trasmissione verso l'archiviazione. I/O Blusam

Il Blusam motore raccoglie tutte le operazioni di aggiornamento eseguite sui record dalle applicazioni e crea lotti di record che vengono inviati per l'elaborazione ai thread dedicati. I lotti vengono quindi memorizzati nell'Blusamarchivio, utilizzando query di aggiornamento in blocco, evitando l'utilizzo di operazioni di persistenza atomica e garantendo il miglior utilizzo possibile della larghezza di banda di rete.

Il meccanismo utilizza un ritardo configurabile (il valore predefinito è un secondo) e una dimensione del lotto configurabile (il valore predefinito è 10000 record). Le query di persistenza della build vengono eseguite non appena viene soddisfatta la prima delle due seguenti condizioni:

- Il ritardo configurato è scaduto e il lotto non è vuoto
- Il numero di record nel lotto da trattare raggiunge il limite configurato

Per informazioni su come configurare il meccanismo write-behind, consulta. the section called "Proprietà facoltative"

Scegliere lo schema di archiviazione corretto

Come mostrato nella sezione precedente, il modo in cui i set di dati vengono archiviati dipende dal loro «peso». Ma cosa è considerato piccolo, medio o grande per un set di dati? Quando scegliere la strategia di archiviazione impaginata anziché quella normale?

La risposta a questa domanda dipende da quanto segue.

- La quantità di memoria disponibile su ciascuno dei server che ospitano le applicazioni modernizzate che utilizzeranno tali set di dati.
- La quantità di memoria disponibile nell'infrastruttura di cache (se presente).

Quando si utilizza uno schema di archiviazione di indici non impaginati, le raccolte complete di indici a chiave e dimensioni dei record verranno caricate nella memoria del server all'ora di apertura del set di dati, per ogni set di dati. Inoltre, se è prevista la memorizzazione nella cache, tutti i record dei set di dati potrebbero essere precaricati nella cache con l'approccio normale, il che potrebbe portare all'esaurimento delle risorse di memoria sul lato dell'infrastruttura della cache.

A seconda del numero di chiavi definite, della lunghezza dei valori delle chiavi, del numero di record e del numero di set di dati aperti contemporaneamente, la quantità di memoria consumata può essere valutata approssimativamente per determinati casi d'uso noti.

Per ulteriori informazioni, consulta the section called "Stima dell'impronta di memoria per un determinato set di dati".

Blusam migrazione

Una volta selezionato lo schema di archiviazione appropriato per un determinato set di dati, lo Blusam storage deve essere popolato mediante la migrazione dei set di dati precedenti.

A tal fine, è necessario utilizzare esportazioni binarie non elaborate dei set di dati legacy, senza che venga utilizzata alcuna conversione dei set di caratteri durante il processo di esportazione. Quando trasferite le esportazioni di set di dati dal sistema precedente, assicuratevi di non danneggiare il formato binario. Ad esempio, applicate la modalità binaria quando utilizzate FTP.

Le esportazioni binarie non elaborate contengono solo i record. Il meccanismo di importazione non necessita delle keys/indexes esportazioni poiché tutte keys/indexes vengono ricalcolate al volo dal meccanismo di importazione.

Una volta che è disponibile l'esportazione binaria di un set di dati, esistono diverse opzioni per migrarlo: Blusam

Nell'ambiente gestito per la modernizzazione AWS del mainframe:

- Importa set di dati utilizzando la funzionalità dedicata. Per informazioni, consulta the section called "Importa set di dati per applicazioni".

or

- Utilizza la funzione di importazione in blocco dei set di dati. Consulta the section called "Data set definition reference" e the section called "Esempio di formato di richiesta di set di dati per VSAM".

or

- Usa uno script groovy per importare set di dati, utilizzando servizi di caricamento dedicati.

> **Note**
>
> Per ora l'importazione di LargeKSD e LargeeSDS su ambienti gestiti di modernizzazione del mainframe è possibile solo utilizzando script groovy.

Su AWS Transform for mainframe Runtime su Amazon EC2:

- Importa il set di dati utilizzando. the section called "BlusamConsole di amministrazione"

or

- Utilizza uno script groovy per importare set di dati, utilizzando servizi di caricamento dedicati.

Importa set di dati utilizzando gli script Groovy

Questa sezione ti aiuterà a scrivere script groovy in cui importare set di dati precedenti. Blusam

Inizia con alcune importazioni obbligatorie:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList; //used for alternate keys if any
```

Dopodiché, per ogni set di dati da importare, il codice si basa sul modello dato:

1. crea o cancella un oggetto della mappa
2. riempi la mappa con le proprietà richieste (questo varia a seconda dei tipi di set di dati, vedi sotto per i dettagli)
3. recupera il servizio di caricamento corretto da utilizzare per il tipo di set di dati nel registro dei servizi
4. esegue il servizio, utilizzando la mappa come argomento

Esistono 5 implementazioni del servizio che possono essere recuperate dal registro dei servizi, utilizzando i seguenti identificatori:

- `"BluesamKSDSFileLoader"`: per KSDS di dimensioni small/medium
- `"BluesamESDSFileLoader"`per ESDS small/medium di dimensioni
- `"BluesamRRDSFileLoader"`: per RRDS
- `"BluesamLargeKSDSFileLoader"`: per KSDS di grandi dimensioni
- `"BluesamLargeESDSFileLoader"`: per ESDS di grandi dimensioni

La scelta tra la versione normale o quella grande del servizio per KSDS/ESDS dipende dalla dimensione dei set di dati e dalla strategia di archiviazione che si desidera applicare. Per informazioni su come scegliere la strategia di archiviazione corretta, consulta. the section called "Scegliere lo schema di archiviazione corretto"

Per importare correttamente il set di dati inBlusam, è necessario fornire le proprietà appropriate al servizio di caricamento.

Proprietà comuni:

- Obbligatorio (per tutti i tipi di set di dati)
  - «BlueSamManager»: il valore previsto è `applicationContext.getBean(BluesamManager.class)`
  - «DataSetName»: nome del set di dati, come stringa
  - "inFilePath": percorso per l'esportazione del set di dati precedente, come stringa
  - «recordLength»: la lunghezza del record fissa o 0 per il set di dati a lunghezza di record variabile, come numero intero
- Facoltativo
  - Non supportato per set di dati di grandi dimensioni:
    - «isAppend»: un flag booleano che indica che l'importazione avviene in modalità di aggiunta (aggiunta di record a un set di dati esistente). Blusam
    - «useCompression»: un flag booleano che indica che la compressione verrà utilizzata per memorizzare i metadati.
  - Solo per set di dati di grandi dimensioni:
    - "indexingPageSizeInMb": la dimensione in megabyte di ogni pagina indice, per ciascuna delle chiavi del set di dati, come numero intero strettamente positivo

Proprietà dipendenti dal tipo di set di dati:

- KSDS/KSDS di grandi dimensioni:
  - mandatory
    - «primaryKey»: la definizione della chiave primaria, utilizzando una chiamata al costruttore. `com.netfective.bluage.gapwalk.bluesam.metadata.Key`
  - opzionale:
    - «AlternateKeys»: una lista (`java.util.List`) di definizioni di chiavi alternative, creata utilizzando `com.netfective.bluage.gapwalk.bluesam.metadata.Key` chiamate al costruttore.
- ESD/ESDS di grandi dimensioni:
  - opzionale:

    - «AlternateKeys»: una lista (`java.util.List`) di definizioni di chiavi alternative, creata utilizzando `com.netfective.bluage.gapwalk.bluesam.metadata.Key` chiamate al costruttore.
- ROSSI:
    - nessuna.

Chiamate del costruttore chiave:

- `new Key(int offset, int length)`: crea un oggetto Key, con determinati attributi chiave (offset e length) e non sono consentiti duplicati. Questa variante deve essere utilizzata per definire una chiave primaria.
- `new Key(boolean allowDuplicates, int offset, int length)`: crea un oggetto Key, con determinati attributi chiave (offset e length) e duplicati che consentono il flag.

I seguenti esempi di Groovy illustrano vari scenari di caricamento.

Caricamento di un KSDS di grandi dimensioni, con due chiavi alternative:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;

// Loading a large KSDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "largeKsdsSample");
map.put("inFilePath", "/work/samples/largeKsdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
ArrayList altKeys = [new Key(true, 10, 8), new Key(false, 0, 9)]
map.put("alternateKeys", altKeys);
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamLargeKSDSFileLoader");
service.runService(map);
```

Caricamento di un record ESDS a lunghezza variabile, senza chiavi alternative:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
```

```
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry

// Loading an ESDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "esdsSample");
map.put("inFilePath", "/work/samples/esdsSampleExport");
map.put("recordLength", 0);
def service = ServiceRegistry.getService("BluesamESDSFileLoader");
service.runService(map);
```

Le esportazioni di set di dati a lunghezza variabile conterranno le informazioni RDW (Record Decriptor Word) obbligatorie per consentire la suddivisione dei record in fase di lettura.

Caricamento di un RRDS a lunghezza di record fissa:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry

// Loading a RRDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "rrdsSample");
map.put("inFilePath", "/work/samples/rrdsSampleExport");
map.put("recordLength", 180);
def service = ServiceRegistry.getService("BluesamRRDSFileLoader");
service.runService(map);
```

Caricamento di set di dati in modalità multischema:

Modalità multischema: in alcuni sistemi legacy, i file VSAM sono organizzati in set di file, che consentono ai programmi di accedere e modificare i dati all'interno di partizioni specificate. I sistemi moderni trattano ogni set di file come uno schema, abilitando un partizionamento dei dati e un controllo degli accessi simili.

Per abilitare la modalità multischema nel `application-main.yml` file, fare riferimento a. the section called "Configurazione di Blusam" In questa modalità, i set di dati possono essere caricati in uno schema specifico utilizzando un contesto condiviso, che è un registro in memoria per le informazioni di runtime. Per caricare un set di dati in uno schema specifico, aggiungete al nome del set di dati il nome dello schema pertinente.

Caricamento di un file KSDS in uno schema specifico per la modalità multischema:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;
import com.netfective.bluage.gapwalk.rt.shared.SharedContext;

// Loading a KSDS into Blusam
def map = [:]
String schema = "schema1";
String datasetName = schema+"|"+"ksdsSample";
SharedContext.get().setCurrentBlusamSchema(schema);
schema = SharedContext.get().getCurrentBlusamSchema();
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", datasetName);
map.put("inFilePath", "/work/samples/ksdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamKSDSFileLoader");
service.runService(map);
```

Caricamento di un file KSDS di grandi dimensioni in uno schema specifico per la modalità multischema:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;
import com.netfective.bluage.gapwalk.rt.shared.SharedContext;

// Loading a Large KSDS into Blusam
def map = [:]
String schema = "schema1";
String datasetName = schema+"|"+"largeKsdsSample";
SharedContext.get().setCurrentBlusamSchema(schema);
schema = SharedContext.get().getCurrentBlusamSchema();
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", datasetName);
map.put("inFilePath", "/work/samples/LargeKsdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
```

```
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamLargeKSDSFileLoader");
service.runService(map);
```

Inoltre, è possibile utilizzare una voce di configurazione (da impostare nel file di `application-main.yml` configurazione) per ottimizzare il processo di importazione:

- `bluesam.fileLoading.commitInterval`: un numero intero strettamente positivo, che definisce l'intervallo di commit per il normale ESDS/KSDS/RRDS meccanismo di importazione. Non si applica alle importazioni di set di dati di grandi dimensioni. Il valore predefinito è 100000.

Configurazione di Blusam

La configurazione Blusam avviene nel file di `application-main.yml` configurazione (o nel file di `application-bac.yml` configurazione per la distribuzione autonoma dell'applicazione Blusam Administration Console -- BAC --).

Blusamdeve essere configurato su due aspetti:

- Blusamconfigurazione dell'accesso allo storage e alle cache
- Blusamconfigurazione del motore

Blusamconfigurazione dell'archiviazione e dell'accesso alle cache

Per informazioni su come configurare l'accesso Blusam allo storage e alle cache utilizzando gestori segreti o origini dati, vedere. the section called “AWS Configurazione Transform for mainframe Runtime”

> **Note**
>
> Per quanto riguarda l'accesso allo Blusam storage, le credenziali utilizzate indicheranno un ruolo di connessione, con i relativi privilegi. Affinché il Blusam motore possa funzionare come previsto, il ruolo di connessione deve avere i seguenti privilegi:

- connettersi al database
- create/delete/alter/tronca tabelle e viste
- seleziona/inserisci/elimina/aggiorna le righe nelle tabelle e nelle viste

- eseguire funzioni o procedure

Blusamconfigurazione del motore

Disabilitazione del supporto Blusam

Innanzitutto, ricordiamo che è possibile disabilitare completamente Blusam il supporto impostando la `bluesam.disabled` proprietà `sutrue`. Un messaggio informativo verrà visualizzato nei log del server all'avvio dell'applicazione per ricordare la Blusam disabilitazione:

```
BLUESAM is disabled. No operations allowed.
```

In tal caso non sono necessarie ulteriori Blusam configurazioni e qualsiasi tentativo di utilizzare le funzionalità Blusam correlate (a livello di codice o tramite chiamate REST) genererà un'interruzione `UnsupportedOperationException` nell'esecuzione del codice Java, con un messaggio di spiegazione pertinente sulla disabilitazione. Blusam

Blusamproprietà del motore

Le proprietà di configurazione Blusam del motore sono raggruppate sotto il prefisso bluesam key:

Proprietà obbligatorie

- `cache`: da valutare con l'implementazione della cache scelta. I valori validi sono:
  - `ehcache`: per l'utilizzo locale di ehcache incorporato. Consultate le limitazioni relative ai casi d'uso riportate sopra.
  - `redis`: per l'utilizzo della cache Redis remota condivisa. Questa è l'opzione preferita per il caso d'uso gestito di AWS Mainframe Modernization.
  - `none`: per disabilitare la memorizzazione nella cache dello storage
- `persistence`: da valutare con pgsql (motore PostgreSQL: versione minima 10.0 — versione consigliata >=14.0
- riferimento all'origine dati: `<persistence engine>.dataSource` punterà alla definizione DataSource per la connessione allo storage, definita altrove nel file di configurazione. Blusam Comunemente viene nominato. `bluesamDs`

> **Note**
>
> Ogni volta che Redis viene utilizzato come meccanismo di cache, sia per i dati che per i blocchi (vedi sotto), è necessario configurare l'accesso alle istanze Redis. Per informazioni dettagliate, vedi the section called "Proprietà della cache Redis disponibili".

Proprietà facoltative

BlusamBlocchi: le proprietà sono precedute da `locks`

- `cache`: l'unico valore utilizzabile è`redis`, per specificare che verrà utilizzato il meccanismo di blocco basato su redis (da utilizzare anche quando la cache di Blusam archiviazione è basata su redis). Se la proprietà è mancante o non è impostata su`redis`, verrà invece utilizzato il meccanismo di blocco in memoria predefinito.
- `lockTimeOut`: un valore intero lungo positivo, che indica il timeout espresso in millisecondi prima che un tentativo di bloccare un elemento già bloccato venga contrassegnato come fallito. L'impostazione predefinita è `500`.
- `locksDeadTime`: un valore intero lungo positivo, che rappresenta il tempo massimo, espresso in millisecondi, in cui un'applicazione può contenere un blocco. I lucchetti vengono automaticamente contrassegnati come scaduti e rilasciati dopo tale periodo di tempo. L'impostazione predefinita è; `1000`
- `locksCheck`: una stringa, utilizzata per definire la strategia di controllo dei blocchi utilizzata dall'attuale gestore dei blocchi, sulla rimozione dei Blusam blocchi scaduti. Da selezionare tra i seguenti valori:
  - `off`: non viene eseguito alcun controllo. Sconsigliato, in quanto potrebbero verificarsi blocchi inutili.
  - `reboot`: i controlli vengono eseguiti al riavvio o all'avvio dell'applicazione. Tutti i lucchetti scaduti vengono rilasciati in quel momento. Questa è l'impostazione predefinita.
  - `timeout`: i controlli vengono eseguiti al riavvio o all'avvio dell'applicazione oppure quando scade un timeout durante un tentativo di bloccare un set di dati. I blocchi scaduti vengono rilasciati immediatamente.

Meccanismo write-behind: le proprietà sono precedute dalla chiave: `write-behind`

- `enabled`: `true` (valore predefinito e consigliato) o`false`, per abilitare o disabilitare il meccanismo write-behind. La disabilitazione del meccanismo avrà un forte impatto sulle prestazioni di scrittura ed è sconsigliata.
- `maxDelay`: una durata massima per l'attivazione dei thread. Il valore predefinito è (un secondo). `"1s"` In genere è consigliabile mantenere il valore predefinito, a meno che condizioni specifiche non richiedano l'ottimizzazione di questo valore. In ogni caso, il valore deve essere mantenuto basso (inferiore a 3 secondi). Il formato per la stringa di ritardo è: `<integer value><optional whitespace><time unit>` dove `<time unit>` deve essere selezionato tra i seguenti valori:
  - `"ns"`: nanosecondi
  - `"µs"`: microsecondi
  - `"ms"`: millisecondi
  - `"s"`: secondi
- `threads`: il numero di thread write-behind dedicati. L'impostazione predefinita è. `5` È necessario regolare questo valore in base alla potenza di calcolo dell'host su cui è in funzione il Blusam motore. Non è importante utilizzare un valore molto più alto, sperando in un aumento delle prestazioni poiché il fattore limitante diventerà la capacità dello storage RDBMS di gestire numerose query in batch simultanee. I valori consigliati sono generalmente compresi tra 4 e 8.
- `batchSize`: un numero intero positivo che rappresenta il numero massimo di record in un lotto che verranno inviati a un thread per il trattamento di massa. Il suo valore deve essere compreso tra 1 e 32767. L'impostazione predefinita è `10000`. L'utilizzo di 1 come valore vanifica lo scopo del meccanismo, che è quello di evitare l'uso di query di aggiornamento atomiche; il valore minimo adatto da utilizzare è circa. `1000`

EhCache Ottimizzazione integrata: le proprietà hanno il prefisso key: `ehcache`

- `resource-pool`:
  - `size`: dimensione della memoria allocata per la cache incorporata, espressa come stringa. Il valore predefinito è `"1024MB"` (1 gigabyte). Da regolare in base alla memoria disponibile della macchina che ospita il Blusam motore e alla dimensione dei set di dati utilizzati dall'applicazione. Il formato della stringa di dimensioni è: `<integer value><optional whitespace><memory unit>` dove `<memory-unit>` deve essere selezionato tra i seguenti valori:
    - `B`: byte
    - `KB`: kilobyte

    - `MB`: megabyte
    - `GB`: gigabyte
    - `TB`: terabyte
  - `heap`: `true` oppure`false`, per indicare se la cache utilizzerà o meno la memoria heap JVM. L'impostazione predefinita è `true` (l'opzione più veloce per le prestazioni della cache, ma l'archiviazione della cache consuma la memoria della memoria RAM JVM on-heap). L'impostazione di questa proprietà su `false` passerà alla memoria Off-Heap, che sarà più lenta a causa degli scambi richiesti con l'heap JVM.
- `timeToLiveMillis`: La durata (in millisecondi) per la quale una voce della cache rimane nella cache prima di essere considerata scaduta e rimossa. Se questa proprietà non è specificata, le voci della cache non scadranno automaticamente per impostazione predefinita.

Proprietà opzionali per set di dati di grandi dimensioni:

Caching localizzato in memoria per indici impaginati:

- `indexesPrefetchWindowSize`: Questa proprietà si applica ai set di dati Blusam Large che utilizzano operazioni abilitate alla cache Redis. Specifica la dimensione massima della cache in memoria (in MB) disponibile per l'archiviazione degli indici impaginati. Il valore predefinito è 0. Questo valore può essere regolato in base alla memoria di sistema disponibile e alla dimensione dei set di dati in fase di elaborazione.

Esempio di frammento di configurazione:

```
dataSource:
  bluesamDs:
    driver-class-name: org.postgresql.Driver
    ...
    ...
bluesam:
  locks:
    lockTimeOut: 700
  cache: ehcache
  persistence: pgsql
  ehcache:
    resource-pool:
      size: 8GB
  write-behind:
```

```
    enabled: true
    threads: 8
    batchsize: 5000
    indexesPrefetchWindowSize: 25
  pgsql:
    dataSource : bluesamDs
```

Proprietà di configurazione multischema

- `multiSchema`: false (valore predefinito) o true, per disabilitare o abilitare la modalità multischema per Blusam - Disponibile a partire dalla versione 4.4.0.
- `pgsql`:
  - `schemas`: Un elenco di nomi di schemi per i quali l'applicazione utilizzerà in modalità multischema. Blusam
  - `fallbackSchema`: il nome dello schema di fallback da utilizzare in modalità multischema. Se non viene trovato un set di dati nel contesto dello schema corrente, questo schema verrà utilizzato per le operazioni Blusam relative a quel set di dati.

Frammento di configurazione di esempio (con la modalità Multischema abilitata per): Blusam

```
dataSource:
  bluesamDs:
    driver-class-name: org.postgresql.Driver
    ...
    ...
bluesam:
  locks:
    lockTimeOut: 700
  cache: ehcache
  persistence: pgsql
  ehcache:
    resource-pool:
      size: 8GB
  write-behind:
    enabled: true
    threads: 8
    batchsize: 5000
  multiSchema: true
  pgsql:
    dataSource : bluesamDs
```

```
    schemas:
      - "schema1"
      - "schema2"
      - "schema3"
    fallbackSchema: schema3
```

> **Note**
>
> Blusamgli schemi di metadati, inclusi gli schemi elencati nel `application-main.yml` file per la modalità Multischema, vengono creati nel Blusam database se non esistono e l'utente dispone di privilegi sufficienti.

BlusamConfigurazione del file DD

Queste opzioni possono essere utilizzate per la configurazione dei Blusam file (definizione dei dati in JCL) in groovy:

- `largeKSDS()`: indica una natura KSDS di grandi dimensioni. Quando viene utilizzato, se il file specificato è un file mancante e deve essere creato (a seconda della modalità OPEN e dell'opzione di file opzionale), verranno creati un file KSDS di grandi dimensioni e la relativa tabella degli indici.

  ```
  .bluesam("TESTFILE").dataset("TESTFILE").largeKSDS().build()
  ```

- `indexPageSize(Integer param)`: specifica la dimensione della pagina (in mb) della tabella degli indici da creare. Applicabile al file con `largeKSDS()` opzione. Il valore di `param` deve essere strettamente positivo. Verrà utilizzato un valore predefinito di 15 per un `param` valore non valido.

  ```
  .bluesam("TESTFILE").dataset("TESTFILE").largeKSDS().indexPageSize(15).build()
  ```

È possibile impostare configurazioni di file predefinite nel `ds-config.yml` file, che si trova in `entities/src/main/resources/ds-config.yml` (o in altre posizioni per i file di configurazione). Queste impostazioni fungono da configurazioni di riserva quando all'unità di esecuzione non viene fornita alcuna configurazione di file specifica. Tutte le opzioni di configurazione dei file Bluesam sono supportate in. `ds-config.yml`

```
TESTFILE:
 provider: Bluesam
```

```
dataset: TESTFILE
largeKSDS: true
indexPageSize: 15
```

BlusamConsole di amministrazione

L'BlusamAdministration Console (BAC) è un'applicazione Web utilizzata per amministrare lo storage. Blusam Per informazioni sul BAC, vedere. the section called "BlusamConsole di amministrazione"

Appendice

Attributi generali dei metadati dei set di dati

Elenco generale degli attributi di serializzazione dei metadati dei set di dati:

- nome (del set di dati)
- tipo (KSDS, LargeKSDS, ESDS, LargeESDS o RRDS)
- flag di riscaldamento della cache (se il set di dati deve essere precaricato nella cache all'avvio del server o meno)
- flag di utilizzo della compressione (se memorizzare i record in formato compresso o raw)
- data di creazione
- data dell'ultima modifica
- contrassegno di registrazione a lunghezza fissa (se i record del set di dati hanno tutti la stessa lunghezza o meno)
- lunghezza del record: significativa solo per una lunghezza di record fissa
- dimensione della pagina (utilizzata per personalizzare le query sql impaginate utilizzate per precaricare la cache quando necessario)
- dimensione (dimensione del set di dati - lunghezza cumulata dei record)
- ultimo offset (offset, ovvero RBA dell'ultimo record aggiunto al set di dati)
- next offset (prossimo offset disponibile per aggiungere un nuovo record al set di dati)
- se significativa, definizione delle chiavi utilizzate dal set di dati; ogni chiave è definita in base al suo tipo (principale o parte della raccolta di chiavi alternative) e tre attributi:
  - offset: posizione nel record del byte iniziale del valore della chiave;

- lunghezza: lunghezza in byte del valore della chiave. Quindi il valore chiave è l'array di byte che è il sottoinsieme del record che inizia dalla posizione `key offset` e finisce nella posizione; `key offset + length - 1`
- flag duplicates allowed: indica se la chiave accetta o meno i duplicati (impostato su true per consentire i duplicati).

Stima dell'impronta di memoria per un determinato set di dati

Per set di dati di piccole e medie dimensioni, i metadati (dimensioni e indici per varie chiavi) verranno caricati completamente in memoria. L'allocazione di risorse adeguate per la macchina che ospita il server utilizzato per eseguire applicazioni modernizzate richiede di calcolare il consumo di memoria indotto dai set di Blusam dati, in particolare per quanto riguarda i metadati. Questa sezione fornisce risposte pratiche agli operatori interessati.

Le formule fornite si applicano solo a set di dati di Blusam piccole e medie dimensioni e non utilizzano la strategia di archiviazione «di grandi dimensioni».

Blusammetadati del set di dati

Per un set di Blusam dati, i metadati sono suddivisi in due parti:

- metadati principali: contiene informazioni globali sul set di dati. L'impronta di memoria di questo può essere considerata trascurabile rispetto ai metadati interni.
- metadati interni: contengono informazioni sulle dimensioni dei record e sugli indici chiave; quando un set di dati non è vuoto, questo è ciò che consuma memoria quando viene caricato nel server delle applicazioni che ospita le applicazioni modernizzate. Le sezioni seguenti descrivono in dettaglio come la memoria consumata cresce con il numero di record.

Calcolo dell'impronta dei metadati interni

Mappa delle dimensioni dei record

Innanzitutto, i metadati interni memorizzano una mappa per contenere la dimensione di ogni record (come numero intero) dato il relativo RBA (indirizzo di byte relativo, memorizzato come numero lungo).

L'impronta di memoria di quella struttura di dati è, in byte: `80 * number of records`

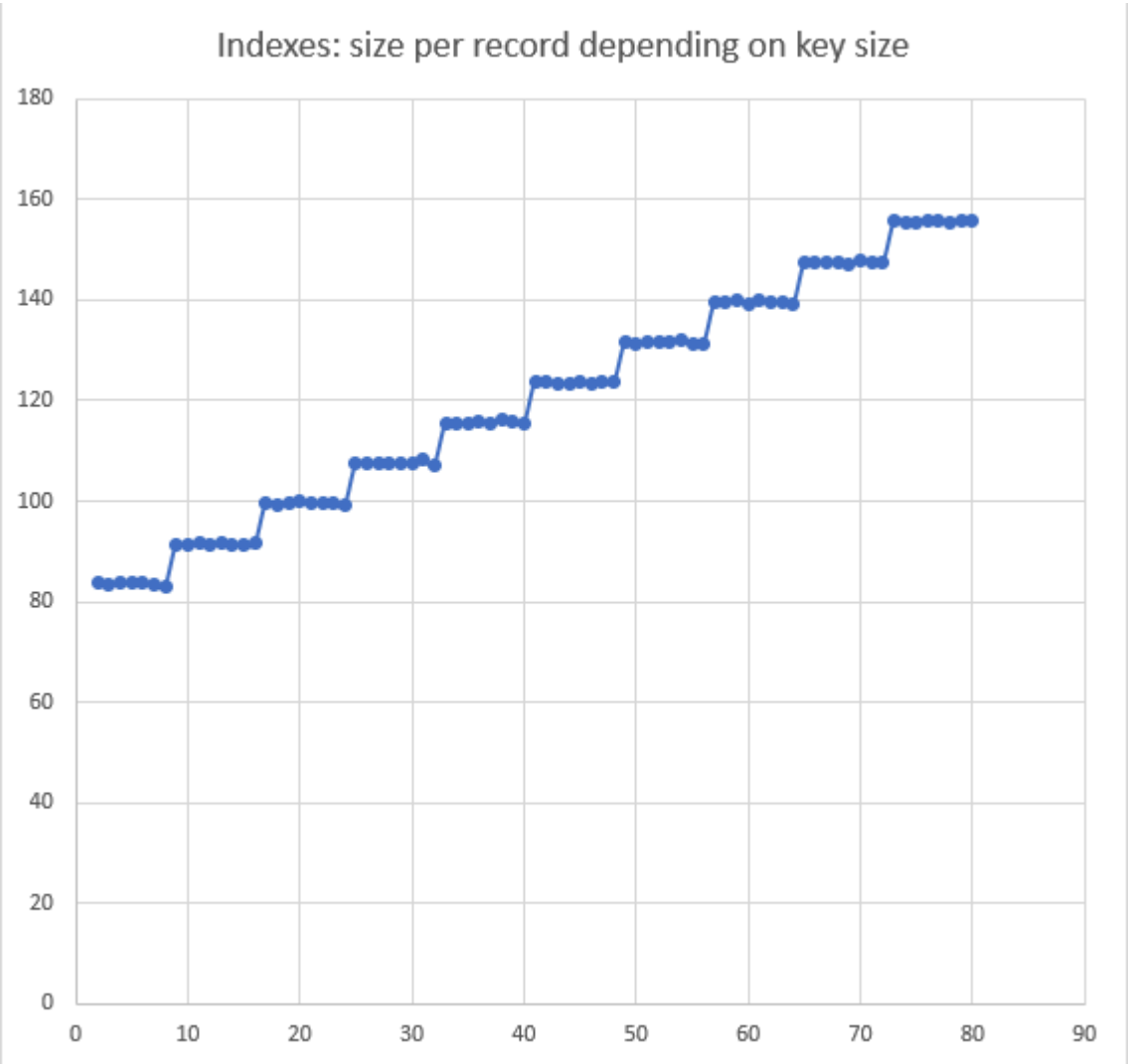
Questo vale per tutti i tipi di set di dati.

Indici

Per quanto riguarda gli indici della chiave primaria di KSDS o delle chiavi alternative sia su ESDS che su KSDS, il calcolo dell'impronta dipende da due fattori:

- il numero di record nel set di dati;
- la dimensione della chiave, in byte.

Il grafico seguente mostra la dimensione dell'indice della chiave per record (asse y) in base alla dimensione della chiave (asse x).



La formula corrispondente per valutare l'impronta per un determinato indice chiave di un set di dati è:

```
index footprint = number of records * ( 83 + 8 (key length / 8))
```

dove '/' sta per la divisione intera.

Esempi:

- set di dati 1:
  - numero di record = 459 996
  - lunghezza della chiave = 15 quindi (lunghezza della chiave/8) = 1
  - impronta dell'indice = 459 996 * (83 + (8*1)) = 41 859 636 byte (= 39 MB circa)
- set di dati 2:
  - numero di record = 13 095 783
  - lunghezza della chiave = 18 quindi (lunghezza della chiave/8) = 2
  - ingombro dell'indice = 13 095 783 * (83 + (8*2)) = 1 296 482.517 byte (= 1,2 GB circa)

L'impronta totale per un determinato set di dati è la somma di tutte le impronte per tutti gli indici delle chiavi e l'impronta per la mappa delle dimensioni dei record.

Ad esempio, prendendo l'esempio del set di dati 2, che ha una sola chiave, l'impronta globale è:

- Mappa delle dimensioni dei record: 13 095 783 * 80 = 1 047 662 640 byte
- Indici chiave: 1 296 482 517 byte (vedi sopra)
- Ingombro totale = 2 344 145 157 byte (= 2,18 GB circa)

AWS Console di Blusam amministrazione Transform for mainframe

L'BlusamAdministration Console (BAC) è un'applicazione Web sicura per la gestione dei set di dati. Blusam Questa guida illustra l'interfaccia utente BAC. Per la gestione remota tramite endpoint REST, vedere. the section called “Blusamendpoint REST della console delle applicazioni”

Argomenti

- Implementazione del BAC
- Utilizzo del BAC
- Formato JSON LISTCAT

Implementazione del BAC

Il BAC è disponibile come singola applicazione web protetta, utilizzando il formato web-archive (.war). È progettato per essere distribuito insieme all'applicazione AWS Transform for mainframe Gapwalk-Application, in un server applicativo Apache Tomcat, ma può anche essere distribuito come applicazione autonoma. Il BAC eredita l'accesso allo storage dalla configurazione Gapwalk-Application, se presente. Blusam

Il BAC dispone di un proprio file di configurazione dedicato, denominato. `application-bac.yml` Per i dettagli sulla configurazione, vederethe section called "File di configurazione dedicato BAC".

Il BAC è protetto. Per informazioni dettagliate sulla configurazione della sicurezza, vedere. the section called "Configurazione della sicurezza per il BAC"

File di configurazione dedicato BAC

Distribuzione autonoma: se il BAC viene distribuito solo con l'applicazione Gapwalk, la connessione allo Blusam storage deve essere configurata nel file di configurazione application-bac.yml.

I valori predefiniti per la configurazione dei set di dati utilizzati per sfogliare i record dei set di dati devono essere impostati nel file di configurazione. Per informazioni, consulta the section called "Navigazione dei record da un set di dati". La pagina di navigazione dei record può utilizzare un meccanismo di maschera opzionale che consente di mostrare una visualizzazione strutturata del contenuto di un record. Alcune proprietà influiscono sulla visualizzazione dei record quando vengono utilizzate le maschere.

Le seguenti proprietà configurabili devono essere impostate nel file di configurazione. L'applicazione BAC non assume alcun valore predefinito per queste proprietà.

Chiave	Tipo	Description
`bac.crud.limit`	intero	Un valore intero positivo che indica il numero massimo di record restituiti durante l'esplorazione dei record. Usare `0` significa illimitato. Valore consigliato: `10` (quindi regola il valore impostato per set di dati nella pagina di

Chiave	Tipo	Description
		navigazione, in base alle tue esigenze).
`bac.crud.encoding`	stringa	Il nome del set di caratteri predefinito, utilizzato per decodificare i byte dei record come contenuto alfanumer ico. Il nome del set di caratteri fornito deve essere compatibi le con Java (consulta la documentazione java per i set di caratteri supportati). Valore consigliato: il set di caratteri legacy utilizzato sulla piattafor ma legacy da cui provengon o i set di dati; questa sarà una variante EBCDIC la maggior parte delle volte.
`bac.crud.initChara cter`	stringa	Il carattere predefinito (byte) utilizzato per avviare gli elementi di dati. È possibile utilizzare due valori speciali: `"LOW-VALUE"` il byte 0x00 (valore consigliato) e `"HI-VALUE"` il byte 0xFF. Utilizzat o quando vengono applicate le maschere.
`bac.crud.defaultCh aracter`	stringa	Il carattere predefinito (byte), come stringa di un carattere, usato per riempire i record (a destra). Valore consigliato: `" "` (spazio). Utilizzato quando vengono applicate maschere.

Chiave	Tipo	Description
`bac.crud.blankCharacter`	stringa	Il carattere predefinito (byte), come stringa di un carattere, utilizzato per rappresentare gli spazi vuoti nei record. Valore consigliato: (spazio). " " Utilizzato quando vengono applicate le maschere.
`bac.crud.strictZoned`	booleano	Un contrassegno per indicare la modalità suddivisa in zone utilizzata per la registrazione. Se`true`, verrà utilizzata la modalità zona rigorosa; se`false`, verrà utilizzata la modalità suddivisa in zone modificata. Valore consigliato:`true`. Usato quando vengono applicate le maschere.
`bac.crud.decimalSeparator`	stringa	Il carattere usato come separatore decimale nei campi numerici modificati (usato quando vengono applicate le maschere).
`bac.crud.currencySign`	stringa	Il carattere predefinito, come stringa di un carattere, utilizzato per rappresentare la valuta nei campi numerici modificati, quando viene applicata la formattazione (utilizzato quando vengono applicate le maschere).

Chiave	Tipo	Description
`bac.crud.pictureCurrencySign`	stringa	Il carattere predefinito, come stringa di un solo carattere, utilizzato per rappresentare la valuta nelle immagini dei campi numerici modificati (utilizzato quando vengono applicate le maschere).

L'esempio seguente è un frammento di file di configurazione.

```
bac.crud.limit: 10
bac.crud.encoding: ascii
bac.crud.initCharacter: "LOW-VALUE"
bac.crud.defaultCharacter: " "
bac.crud.blankCharacter: " "
bac.crud.strictZoned: true
bac.crud.decimalSeparator: "."
bac.crud.currencySign: "$"
bac.crud.pictureCurrencySign: "$"
```

Configurazione della sicurezza per il BAC

La configurazione della sicurezza per il BAC si basa sui meccanismi descritti in questa pagina di documentazione. Lo schema di autenticazione è e OAuth2 vengono forniti i dettagli di configurazione per Amazon Cognito o Keycloak.

Sebbene sia possibile applicare la configurazione generale, alcune specifiche sul BAC devono essere dettagliate qui. L'accesso alle funzionalità BAC è protetto mediante una politica basata sui ruoli e si basa sui seguenti ruoli.

- RUOLO_UTENTE:
  - Ruolo utente di base
  - Non è consentita l'importazione, l'esportazione, la creazione o l'eliminazione di set di dati
  - Nessun controllo sulle politiche di memorizzazione nella cache
  - Non sono consentite funzionalità di amministrazione
- ROLE_ADMIN:

- Eredita le autorizzazioni ROLE_USER
- Sono consentite tutte le operazioni sui set di dati
- Amministrazione delle politiche di memorizzazione nella cache consentita

Installazione delle maschere

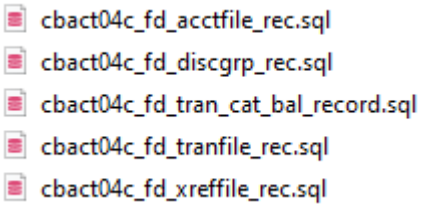
Nell'Blusamarchiviazione, i record dei set di dati vengono archiviati in una colonna di matrice di byte nel database, per considerazioni sulla versatilità e sulle prestazioni. Avere accesso a una visualizzazione strutturata, utilizzando campi, dei record aziendali, basata sul punto di vista dell'applicazione è una comoda funzionalità del BAC. Ciò si basa sulle maschere SQL prodotte durante il processo di modernizzazione basato su AWS Transform for mainframe.

Per generare le maschere SQL, assicurati di impostare l'opzione pertinente (`export.SQL.masks`) nella configurazione di AWS Transform for mainframe refactor Transformation Center su true:



Le maschere fanno parte degli artefatti di modernizzazione che possono essere scaricati da AWS Transform for mainframe refactor per un determinato progetto. Sono script SQL, organizzati da programmi modernizzati, che forniscono il punto di vista applicativo sui record dei set di dati.

Ad esempio, utilizzando l'applicazione di CardDemo esempio AWS, puoi trovare negli artefatti scaricati dal risultato della modernizzazione di questa applicazione, le seguenti maschere SQL per il programma c.CBL: CBACT04



Ogni nome di maschera SQL è la concatenazione del nome del programma e del nome della struttura del record per un determinato set di dati all'interno del programma.

Ad esempio, osservando il programma [CBACT04c.CBL], la voce di controllo del file specificata:

```
    FILE-CONTROL.
        SELECT TCATBAL-FILE ASSIGN TO TCATBALF
               ORGANIZATION IS INDEXED
               ACCESS MODE  IS SEQUENTIAL
               RECORD KEY   IS FD-TRAN-CAT-KEY
               FILE STATUS  IS TCATBALF-STATUS.
```

è associato alla definizione del record FD specificata

```
       FILE SECTION.
       FD  TCATBAL-FILE.
       01  FD-TRAN-CAT-BAL-RECORD.
           05 FD-TRAN-CAT-KEY.
              10 FD-TRANCAT-ACCT-ID             PIC 9(11).
              10 FD-TRANCAT-TYPE-CD             PIC X(02).
              10 FD-TRANCAT-CD                  PIC 9(04).
           05 FD-FD-TRAN-CAT-DATA               PIC X(33).
```

La maschera SQL corrispondente denominata `cbact04c_fd_tran_cat_bal_record.SQL` è la maschera che fornisce il punto di vista del programma CBACT04 c.CBL sul record FD denominato. `FD-TRAN-CAT-BAL-RECORD`

Il suo contenuto è:

```
-- Generated by AWS Transform for mainframe Velocity
-- Mask : cbact04c_fd_tran_cat_bal_record

INSERT INTO mask (name, length) VALUES ('cbact04c_fd_tran_cat_bal_record', 50);
```

```
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_trancat_acct_id', 1, 11, false, 'zoned', 'integerSize=11!fractionalSize=0!
signed=false', (SELECT MAX(id) FROM mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_trancat_type_cd', 12, 2, false, 'alphanumeric', 'length=2', (SELECT MAX(id) FROM
 mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk)
 VALUES ('fd_trancat_cd', 14, 4, false, 'zoned', 'integerSize=4!fractionalSize=0!
signed=false', (SELECT MAX(id) FROM mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_fd_tran_cat_data', 18, 33, false, 'alphanumeric', 'length=33', (SELECT MAX(id)
 FROM mask));
```

Le maschere vengono archiviate nella Blusam memoria utilizzando due tabelle:

- maschera: utilizzata per identificare le maschere. Le colonne della tabella mas sono:
  - nome: utilizzato per memorizzare l'identificazione della maschera (usato come chiave primaria, quindi deve essere univoco)
  - lunghezza: dimensione in byte della maschera di registrazione
- mask_item: usato per memorizzare i dettagli della maschera. Ogni campo elementare di una definizione di record FD produrrà una riga nella tabella mask_item, con dettagli su come interpretare la parte del record specificata. Le colonne della tabella mask_item sono:
  - nome: nome del campo di registrazione, basato sul nome elementare, utilizzando lettere minuscole e sostituendo il trattino con il trattino basso
  - c_offset: offset a base 1 della sottoparte del record, utilizzato per il contenuto del campo
  - length: lunghezza in byte della sottoparte del record, utilizzata per il contenuto del campo
  - skip: contrassegno per indicare se la parte del record specificata deve essere saltata o meno, nella presentazione della vista
  - type: il tipo di campo (basato sulla precedente clausola picture)
  - opzioni: opzioni di tipo aggiuntive, dipendenti dal tipo
  - mask_fk: riferimento all'identificatore della maschera a cui allegare questo elemento

Tenere presente quanto segue:

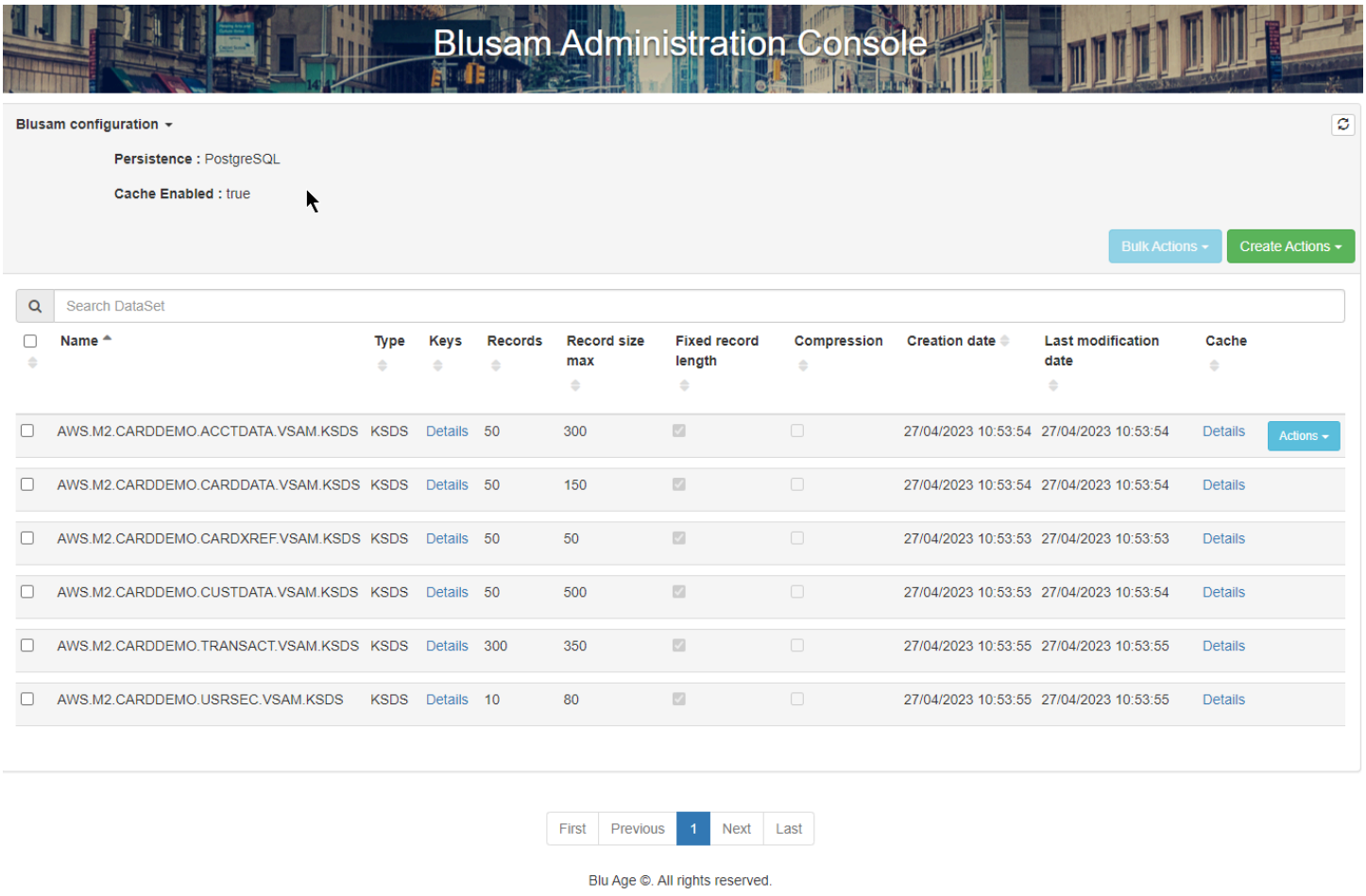
- Le maschere SQL rappresentano il punto di vista di un programma sui record di un set di dati: diversi programmi potrebbero avere un punto di vista diverso su un determinato set di dati; installa solo le maschere che ritieni pertinenti allo scopo.

- Una maschera SQL può anche rappresentare il punto di vista di un programma basato su una struttura di dati 01 della sezione WORKING STORAGE, non solo da un record FD. Le maschere SQL sono organizzate in sottocartelle in base alla loro natura:
  - Le maschere basate su record FD si troveranno nella sottocartella denominata `file`
  - 01 maschere basate sulla struttura dei dati si troveranno nella sottocartella denominata `working`

  Sebbene le definizioni dei record FD corrispondano sempre al contenuto del record di un set di dati, le strutture di dati 01 potrebbero non essere allineate o rappresentare solo un sottoinsieme di un record del set di dati. Prima di utilizzarli, ispeziona il codice e comprendi le possibili carenze.

Utilizzo del BAC

Poiché il BAC è protetto e fornisce le autorizzazioni per utilizzare le funzionalità in base al ruolo dell'utente, il primo passaggio per accedere all'applicazione consiste nell'autenticarsi. Dopo la fase di autenticazione, verrai reindirizzato alla home page. La home page presenta l'elenco impaginato dei set di dati presenti nella memoria: Blusam



Per tornare alla home page con l'elenco dei set di dati, scegliete il AWS Transform for mainframe logo nell'angolo superiore sinistro di qualsiasi pagina dell'applicazione. L'immagine seguente mostra il logo.

L'intestazione pieghevole, denominata "Blusamconfigurazione», contiene informazioni sulla configurazione di archiviazione utilizzata: Blusam

- `Persistence`: il motore di archiviazione persistente (PostgreSQL)
- `Cache Enabled`: se la cache di archiviazione è abilitata

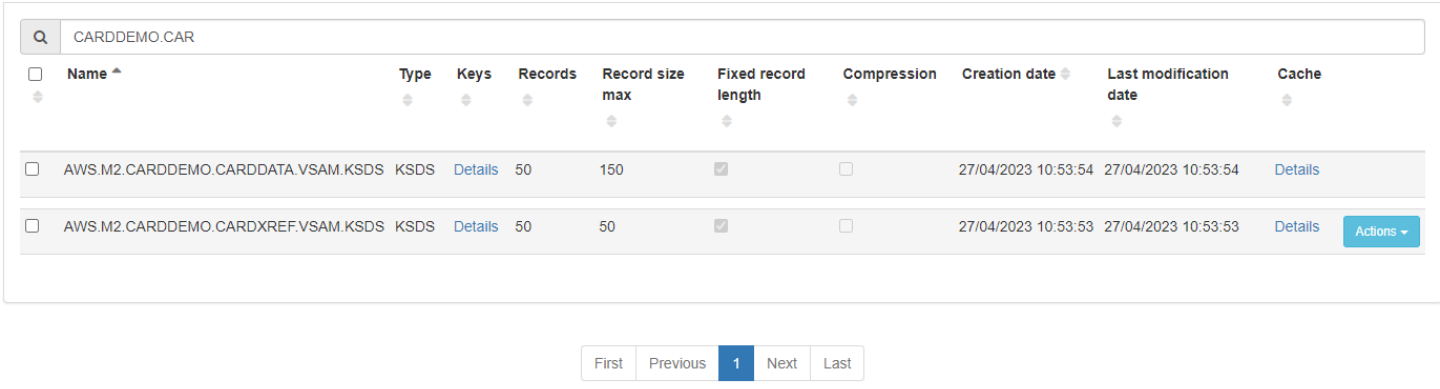
Sul lato destro dell'intestazione, due elenchi a discesa, ognuno dei quali elenca le operazioni relative ai set di dati:

- Azioni in blocco
- Crea azioni

Per ulteriori informazioni sui contenuti dettagliati di questi elenchi, vederethe section called "Operazioni esistenti sui set di dati".

Il pulsante Azioni in blocco è disabilitato quando non è stata effettuata alcuna selezione di set di dati.

Puoi utilizzare il campo di ricerca per filtrare l'elenco in base ai nomi dei set di dati:



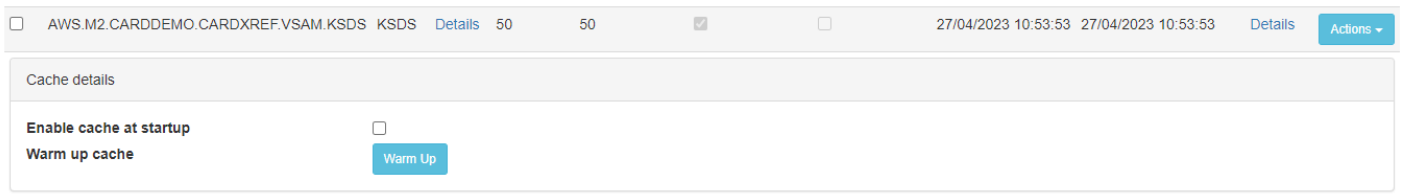
L'elenco impaginato che segue mostra un set di dati per riga della tabella, con le seguenti colonne:

- Casella di selezione: una casella di controllo per selezionare il set di dati corrente.
- Nome: il nome del set di dati.

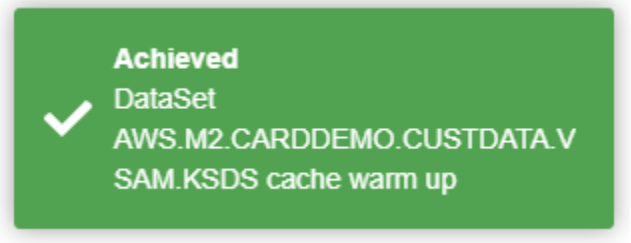
- Tipo: il tipo di set di dati, uno dei seguenti:
  - KSDS
  - EDDS
  - ROSSI
- Chiavi: un link per mostrare o nascondere i dettagli sulle chiavi (se presenti). Ad esempio, il KSDS specificato ha la chiave primaria obbligatoria e una chiave alternativa.



  C'è una riga per chiave, con le seguenti colonne. Nessuno dei campi è modificabile.
  - Natura della chiave: chiave primaria o chiave alternativa
  - Nome: il nome della chiave
  - Unico: se la chiave accetta voci duplicate
  - Offset: offset dell'inizio dei tasti all'interno del record
  - Lunghezza: lunghezza in byte della parte chiave del record
- Record: il numero totale di record nel set di dati.
- Dimensione massima del record: la dimensione massima dei record, espressa in byte.
- Lunghezza di record fissa: una casella di controllo che indica se i record sono a lunghezza fissa (selezionato) o a lunghezza variabile (non selezionato).
- Compressione: una casella di controllo che indica se la compressione viene applicata (selezionata) o meno (non selezionata) agli indici memorizzati.
- Data di creazione: la data in cui il set di dati è stato creato nello storage. Blusam
- Data ultima modifica: la data dell'ultimo aggiornamento del set di dati nello Blusam storage.
- Cache: un link per mostrare o nascondere i dettagli sulla strategia di memorizzazione nella cache applicata a questo set di dati.



- Abilita la cache all'avvio: una casella di controllo per specificare la strategia di memorizzazione nella cache di avvio per questo set di dati. Se selezionato, il set di dati verrà caricato nella cache al momento dell'avvio.
- Warm up cache: un pulsante per caricare il set di dati specificato nella cache, a partire immediatamente (ma l'idratazione della cache richiede del tempo, a seconda della dimensione del set di dati e del numero di chiavi). Dopo che il set di dati è stato caricato nella cache, viene visualizzata una notifica come la seguente.

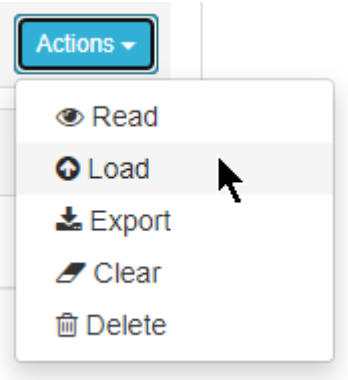


- Azioni: un elenco a discesa delle possibili operazioni sui set di dati. Per informazioni dettagliate, vedi the section called "Operazioni esistenti sui set di dati".

Nella parte inferiore della pagina, c'è un normale widget di navigazione impaginato per sfogliare le pagine dell'elenco dei set di dati.

Operazioni esistenti sui set di dati

Per ogni set di dati nell'elenco impaginato, è disponibile un elenco a discesa Azioni con il seguente contenuto:



Ogni elemento dell'elenco è un collegamento attivo che consente di eseguire l'azione specificata sul set di dati:

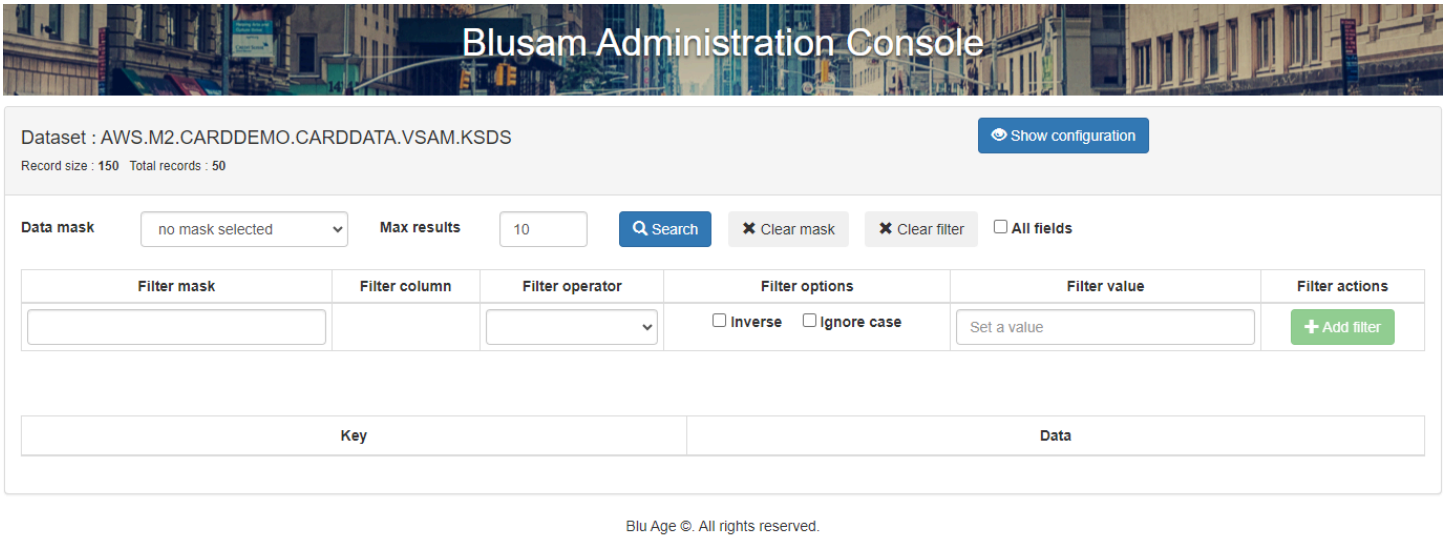
- Leggi: sfoglia i record dei set di dati
- Carica: importa i record da un file di set di dati precedente

- Esporta: esporta i record in un file flat (compatibile con i sistemi legacy)
- Cancella: rimuove tutti i record dal set di dati
- Elimina: rimuove il set di dati dalla memoria

I dettagli per ciascuna azione sono forniti nelle sezioni seguenti.

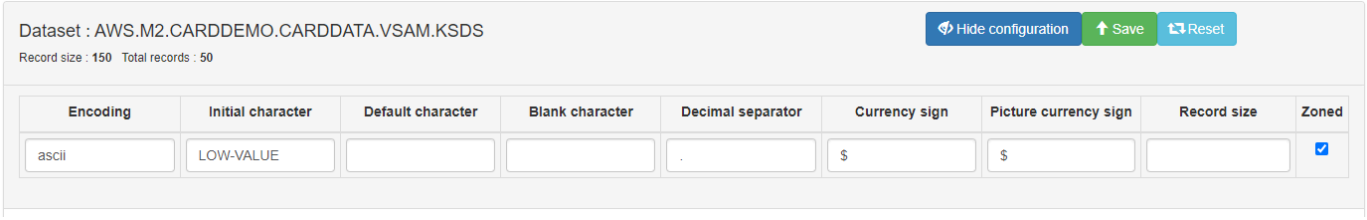
Navigazione dei record da un set di dati

Quando si sceglie l'azione Leggi per un determinato set di dati, viene visualizzata la pagina seguente.



La pagina è composta da:

- un'intestazione, con:
  - Set di dati: il nome del set di dati
  - Dimensione del record: la lunghezza fissa del record, espressa in byte
  - Record totali: il numero totale di record archiviati per questo set di dati
  - Mostra pulsante di configurazione (sul lato destro): un pulsante per attivare la configurazione show/hide del set di dati. Inizialmente, la configurazione è nascosta. Quando si utilizza il pulsante, viene visualizzata la configurazione, come mostrato nell'immagine seguente.

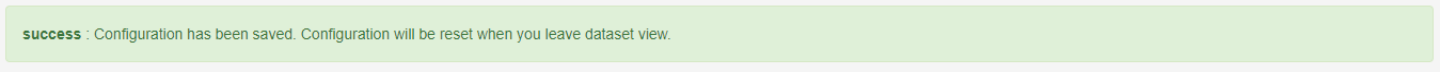


Quando viene mostrata la configurazione, due nuovi pulsanti: Salva e Ripristina, utilizzati rispettivamente per:

- salva la configurazione per questo set di dati e la sessione di lavoro corrente
- reimposta la configurazione ai valori predefiniti per tutti i campi.

- Un elenco di proprietà configurabili per personalizzare l'esperienza di navigazione per un determinato set di dati.

Le proprietà configurabili corrispondono alle proprietà di configurazione descritte in. the section called "File di configurazione dedicato BAC" Fai riferimento a quella sezione per comprendere il significato di ogni colonna e dei valori applicabili. Ogni valore può essere ridefinito qui per il set di dati e salvato per la sessione di lavoro (utilizzando il pulsante Salva). Dopo aver salvato la configurazione, viene visualizzato un banner simile a quello mostrato nell'immagine seguente.



Il banner indica che la sessione di lavoro termina quando si esce dalla pagina corrente.

Esiste una proprietà configurabile aggiuntiva che non è documentata nella sezione di configurazione: Dimensione del record. Viene utilizzato per specificare una determinata dimensione del record, espressa in byte, che filtrerà le maschere applicabili a questo set di dati: solo le maschere la cui lunghezza totale corrisponde alla dimensione del record specificata verranno elencate nell'elenco a discesa Maschera di dati.

Il recupero dei record dal set di dati viene attivato dal pulsante Cerca, utilizzando tutte le opzioni e i filtri nelle vicinanze.

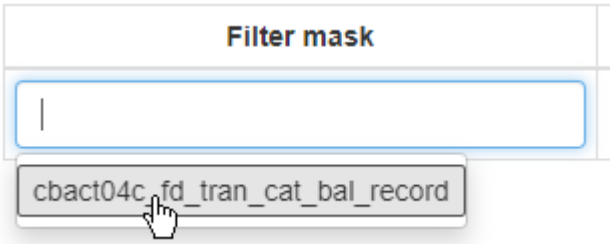
Prima riga di opzioni:

- l'elenco a discesa Data mask mostra le maschere applicabili (rispettando la dimensione del record). Tieni presente che la corrispondenza delle dimensioni del record non è sufficiente per essere una maschera applicabile efficace. La definizione della maschera deve inoltre essere compatibile con il contenuto dei record. La maschera di dati selezionata qui ha
- Numero massimo di risultati: limita il numero di record recuperati dalla ricerca. Imposta su 0 per risultati illimitati (impaginati) dal set di dati.
- Pulsante di ricerca: avvia il recupero dei record utilizzando filtri e opzioni

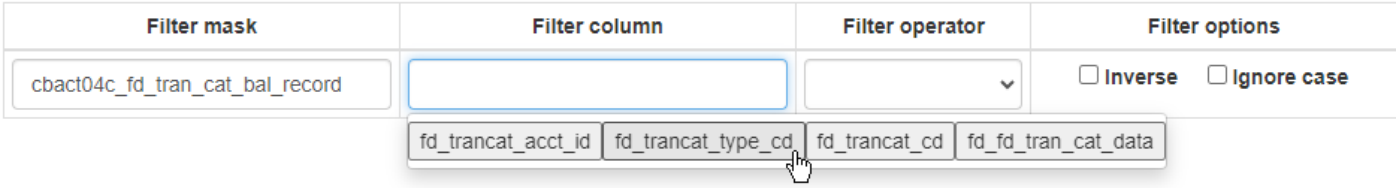
- Pulsante Cancella maschera: cancellerà l'eventuale maschera usata e riporterà la pagina dei risultati a una presentazione non key/data elaborata.
- Pulsante Cancella filtro: cancellerà i filtri usati, se presenti, e aggiornerà la pagina dei risultati di conseguenza.
- Tutti i campi si attivano: se selezionati, gli elementi della maschera definiti con `skip = true` vengono comunque mostrati, altrimenti gli elementi della maschera con `skip = true` sono nascosti.

Prossime righe di filtri: È possibile definire un elenco di filtri, in base all'utilizzo delle condizioni di filtraggio applicate ai campi (colonne) di una determinata maschera, come mostrato nell'immagine seguente.

- Maschera di filtro: il nome della maschera da cui scegliere la colonna di filtraggio. Quando si sceglie il campo, viene visualizzato l'elenco delle maschere applicabili. Puoi scegliere la maschera che desideri da quell'elenco.



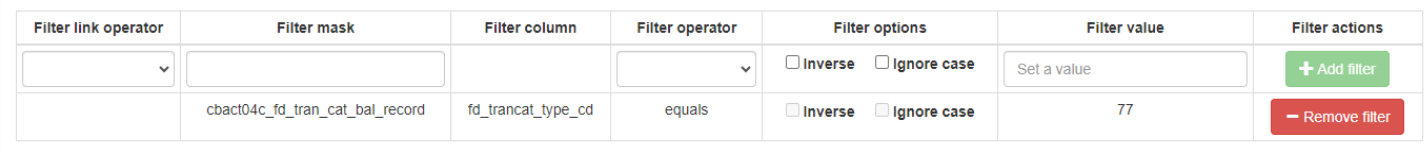
- Colonna di filtro: il nome del campo (colonna) della maschera, usato per filtrare i record. Quando si sceglie il campo, viene visualizzato l'elenco delle colonne della maschera. Per riempire il campo della colonna Filtro, scegli la cella desiderata.



- Operatore di filtro: un operatore da applicare alla colonna selezionata. Sono disponibili i seguenti operatori.
  - uguale a: il valore della colonna per il record deve essere uguale al valore del filtro
  - inizia con: il valore della colonna per il record deve iniziare con il valore del filtro
  - termina con: il valore della colonna per il record deve terminare con il valore del filtro
  - contiene: il valore della colonna per il record deve contenere il valore del filtro

- Opzioni di filtro:
  - Inverso: applica la condizione inversa per l'operatore del filtro; ad esempio, «è uguale a» è sostituito da «non è uguale a»;
  - Ignora maiuscole/minuscole: ignora le maiuscole nei confronti alfanumerici per l'operatore di filtro
- Valore del filtro: il valore utilizzato per il confronto dall'operatore di filtro con la colonna del filtro.

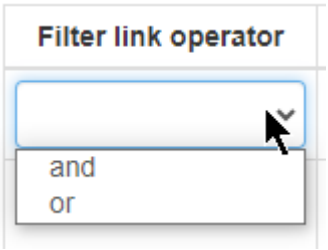
Una volta impostato il numero minimo di elementi del filtro (almeno: è necessario impostare la maschera di filtro, la colonna del filtro, l'operatore di filtro e il valore del filtro), il pulsante Aggiungi filtro è abilitato e facendo clic su di esso si crea una nuova condizione di filtro sui record recuperati. Un'altra riga vuota della condizione di filtro viene aggiunta in alto e la condizione di filtro aggiunta presenta un pulsante Rimuovi filtro che può essere utilizzato per sopprimere la condizione di filtro specificata:



Quando avvii la ricerca, i risultati filtrati vengono visualizzati in una tabella impaginata.

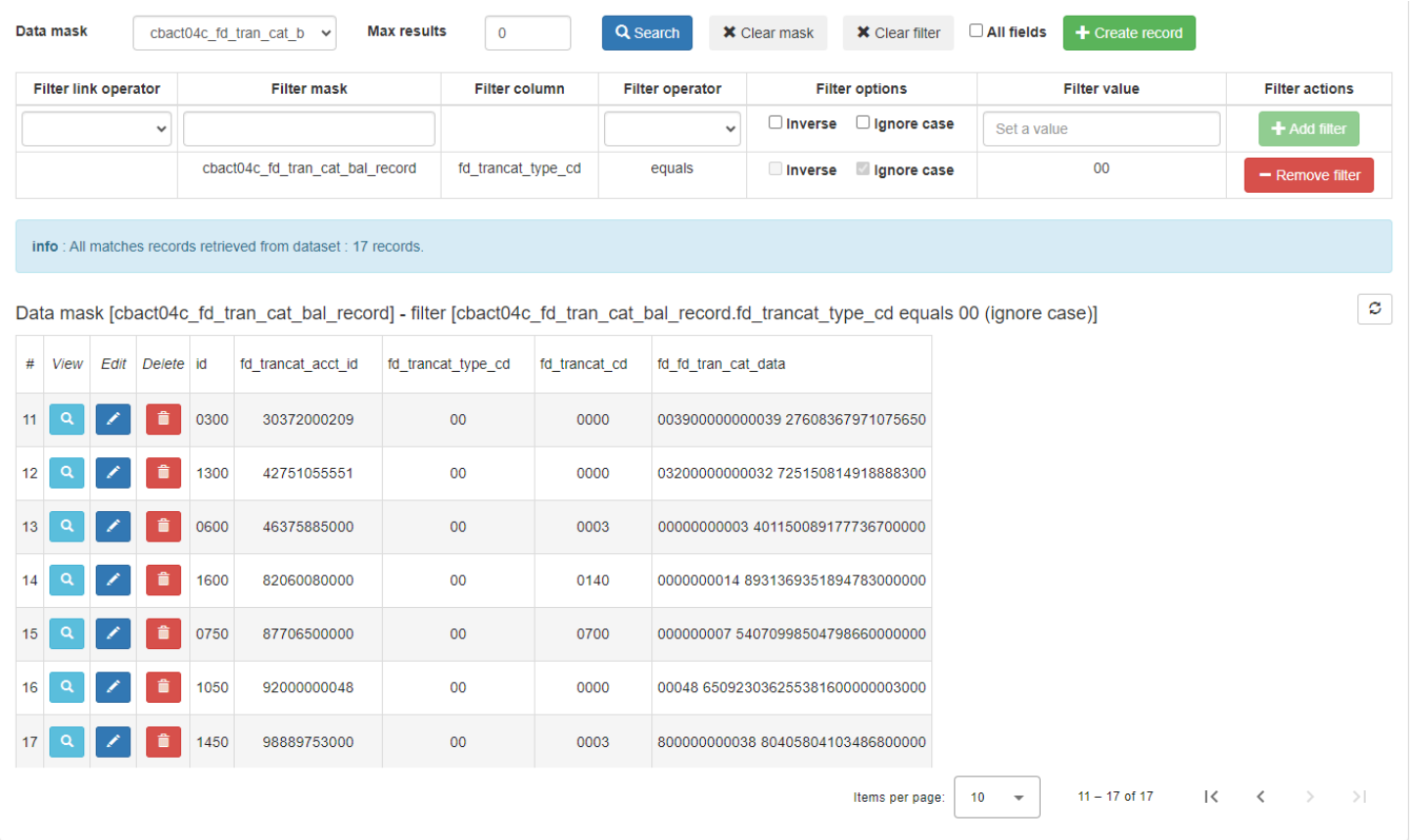
Nota

- I filtri successivi sono collegati da un e o. Ogni nuova definizione di filtro inizia impostando l'operatore di collegamento, come mostrato nell'immagine seguente.



- È possibile che non vi siano record che soddisfino le condizioni di filtro specificate.

Altrimenti, la tabella dei risultati è simile a quella nell'immagine seguente.



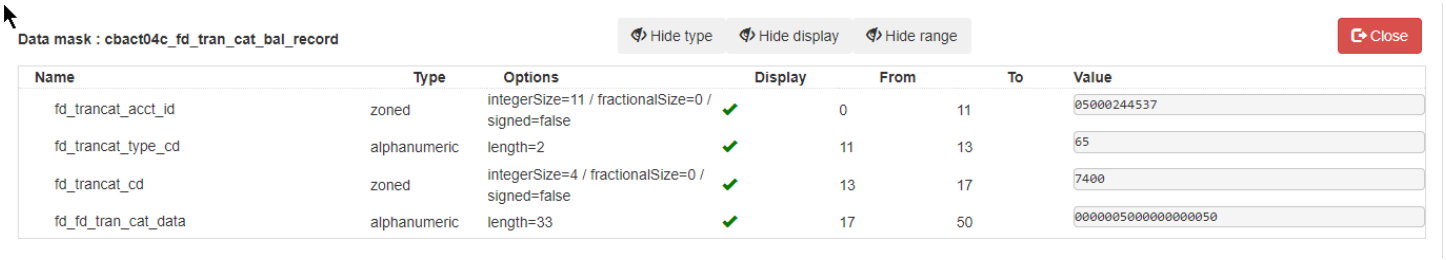
Un'intestazione indica il numero totale di record che soddisfano le condizioni del filtro. Dopo l'intestazione, viene visualizzato quanto segue.

- Promemoria della maschera di dati utilizzata (se presente) e delle condizioni del filtro.
- Un pulsante di aggiornamento che puoi utilizzare per attivare l'aggiornamento dell'intera tabella dei risultati con gli ultimi valori della Blusam memoria (poiché potrebbe essere stata aggiornata da un altro utente, ad esempio).

Per ogni record recuperato, la tabella ha una riga che mostra il risultato dell'applicazione della maschera di dati al contenuto dei record. Ogni colonna è l'interpretazione della sottoparte del record in base al tipo di colonna (e utilizzando la codifica selezionata). A sinistra di ogni riga, ci sono tre pulsanti:

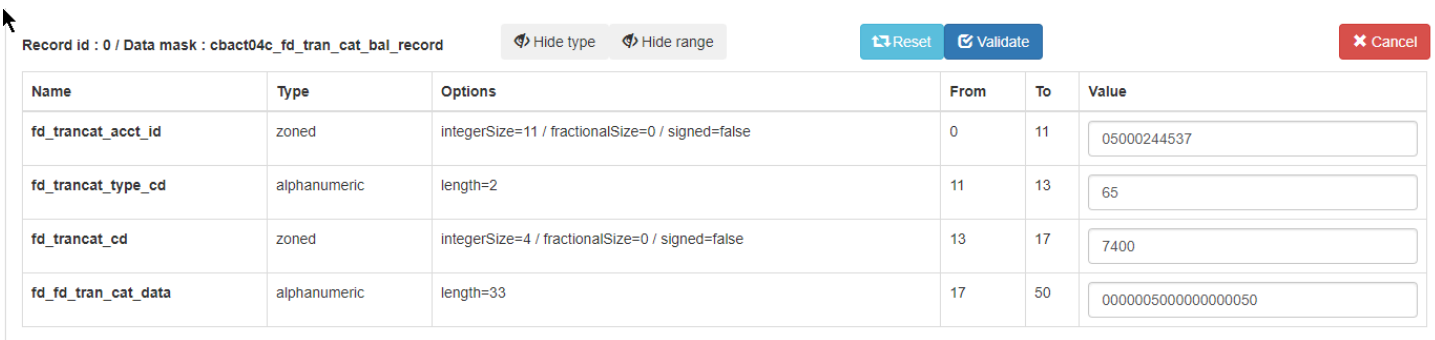
- un pulsante con lente di ingrandimento: conduce a una pagina dedicata che mostra il contenuto dettagliato del record
- un pulsante penna: conduce a una pagina di modifica dedicata ai contenuti del record:
- un pulsante cestino: usato per cancellare il dato record dallo storage blusam

Visualizzazione dettagliata del contenuto del record:



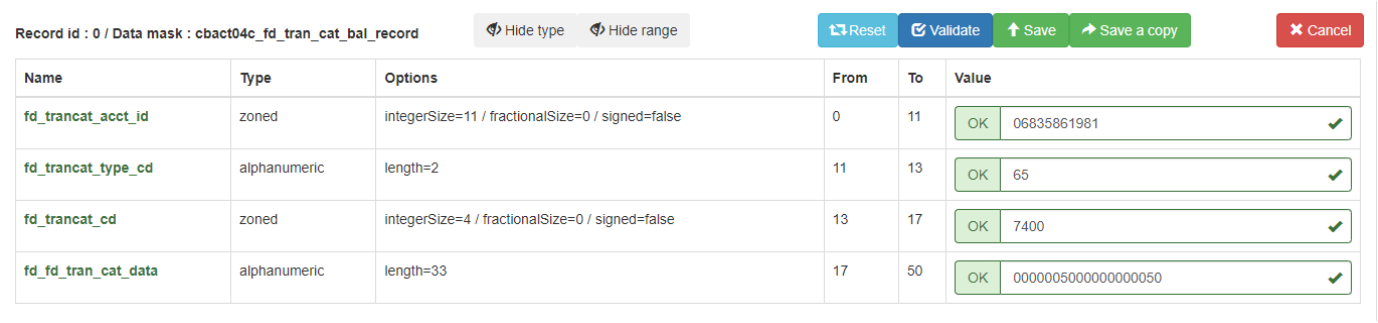
- Tre pulsanti per nascondere o mostrare alcune colonne:
  - Mostra/nasconde il tipo
  - Mostra/nasconde la bandiera del display
  - Mostra/nasconde l'intervallo
- Per uscire da questa pagina dedicata e tornare alla tabella dei risultati, scegli Chiudi.
- Ogni riga rappresenta una colonna della maschera di dati, con le seguenti colonne:
  - Nome: il nome della colonna
  - Tipo: il tipo di colonna
  - Display: l'indicatore del display; verrà visualizzato un segno di spunta verde se l'elemento della maschera corrispondente è definito con`skip = false`, altrimenti verrà visualizzata una croce rossa
  - From & To: l'intervallo basato su 0 per la sottoparte del record
  - Valore: il valore interpretato della sottoparte del record, utilizzando il tipo e la codifica

Modifica del contenuto del record:

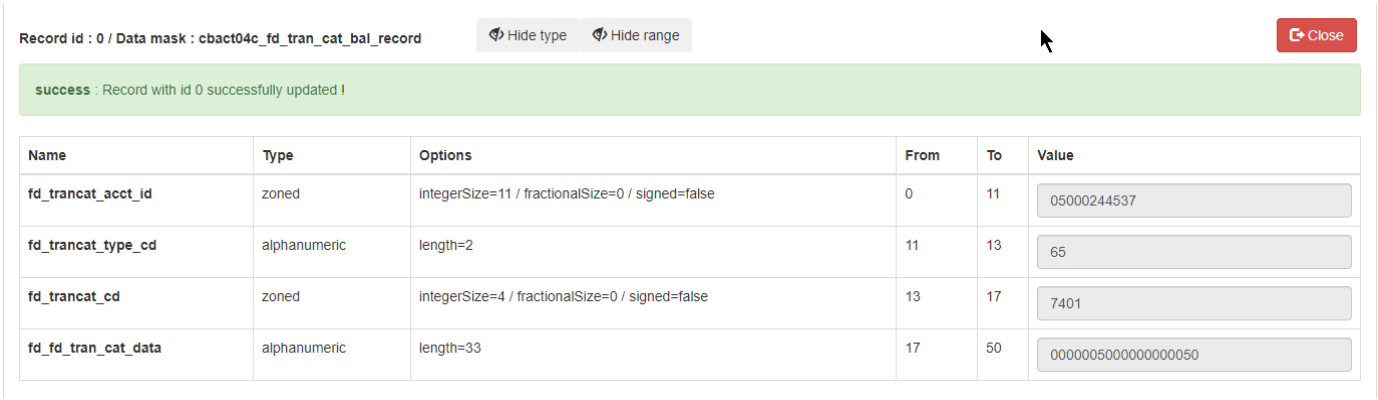


La pagina di modifica è simile alla pagina di visualizzazione descritta sopra, tranne per il fatto che i valori degli elementi della maschera sono modificabili. Tre pulsanti controllano il processo di aggiornamento:

- Ripristina: ripristina i valori modificabili ai valori dei record iniziali (precedenti a qualsiasi edizione);
- Convalida: convalida l'input, per quanto riguarda il tipo di elemento della maschera. Per ogni elemento della maschera, il risultato della convalida verrà stampato utilizzando etichette visive (0Ke una casella di controllo se la convalida è riuscita, e una croce rossa se la convalida non è riuscita, ERR0R insieme a un messaggio di errore che fornisce suggerimenti sull'errore di convalida). Se la convalida ha avuto esito positivo, verranno visualizzati due nuovi pulsanti:
  - Salva: tenta di aggiornare il record esistente nella memoria Blusam
  - Salva una copia: tenta di creare un nuovo record nella Blusam memoria



  - Se il salvataggio del record nella memoria ha esito positivo, viene visualizzato un messaggio e la pagina passerà alla modalità di sola lettura (i valori degli elementi della maschera non possono più essere modificati):



  - Se per qualsiasi motivo la persistenza del record nella memoria non riesce, viene visualizzato un messaggio di errore in rosso che indica il motivo dell'errore. Il caso più comune di errore è rappresentato dal fatto che la memorizzazione del record comporterebbe il danneggiamento della chiave (chiave non valida o duplicata). Per un'illustrazione, vedere la nota seguente.
  - Per uscire, scegliete il pulsante Chiudi.
- Annulla: termina la sessione di modifica, chiude la pagina e torna alla pagina con l'elenco dei record.

Nota:

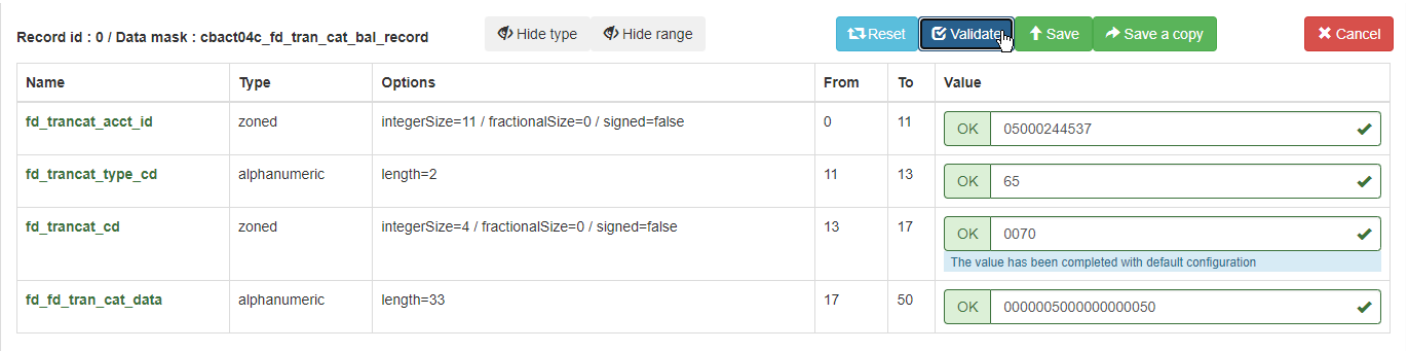
- Il meccanismo di convalida verifica solo che il valore dell'elemento maschera sia formalmente compatibile con il tipo di elemento della maschera. Ad esempio, vedi questa convalida fallita su un elemento di maschera numerica:



- Il meccanismo di convalida potrebbe tentare di correggere automaticamente un input non valido, visualizzando un messaggio informativo in blu per indicare che il valore è stato corretto automaticamente, in base al tipo. Ad esempio, inserendo 7 XX0 come valore numerico nell'elemento della maschera numerica: `fd_trncat_cd`

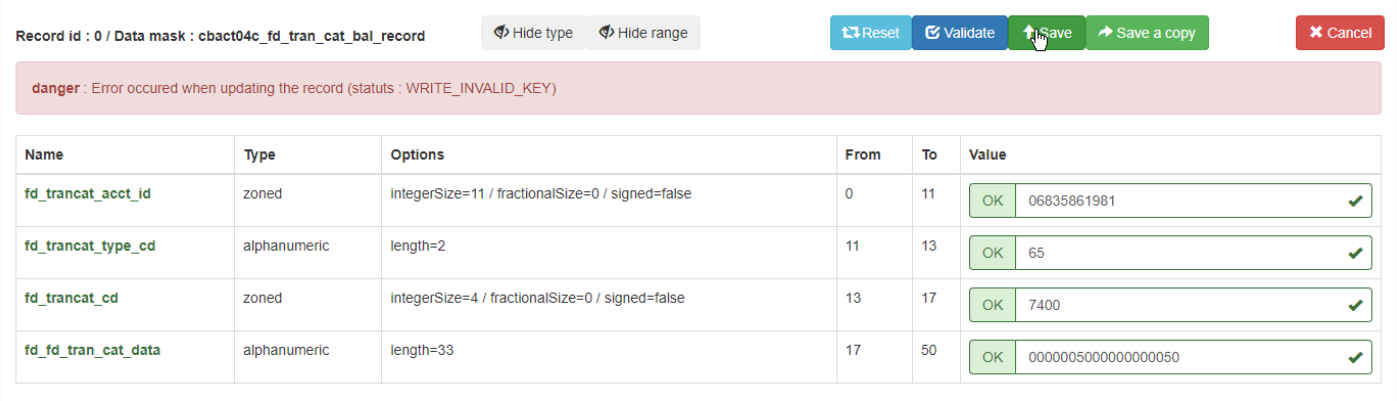


  La convalida della chiamata porta a quanto segue:



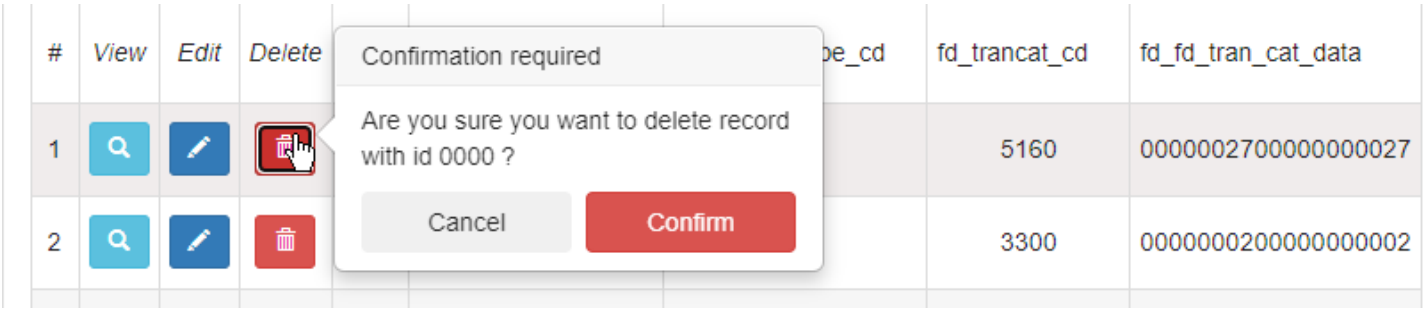
- Il meccanismo di convalida non verifica se il valore dato è valido in termini di integrità della chiave (se è coinvolta una chiave unica per il dato set di dati). Ad esempio, nonostante la convalida

abbia esito positivo, se i valori forniti portano a una situazione di chiave non valida o duplicata, la persistenza avrà esito negativo e verrà visualizzato un messaggio di errore:



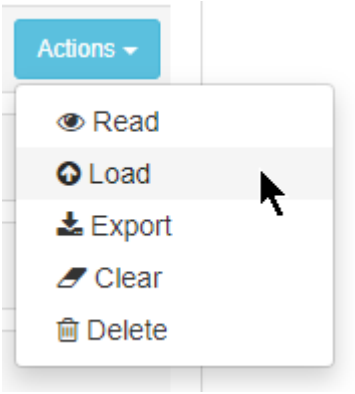
Eliminazione di un record:

Per eliminare un record, scegli il pulsante del cestino:

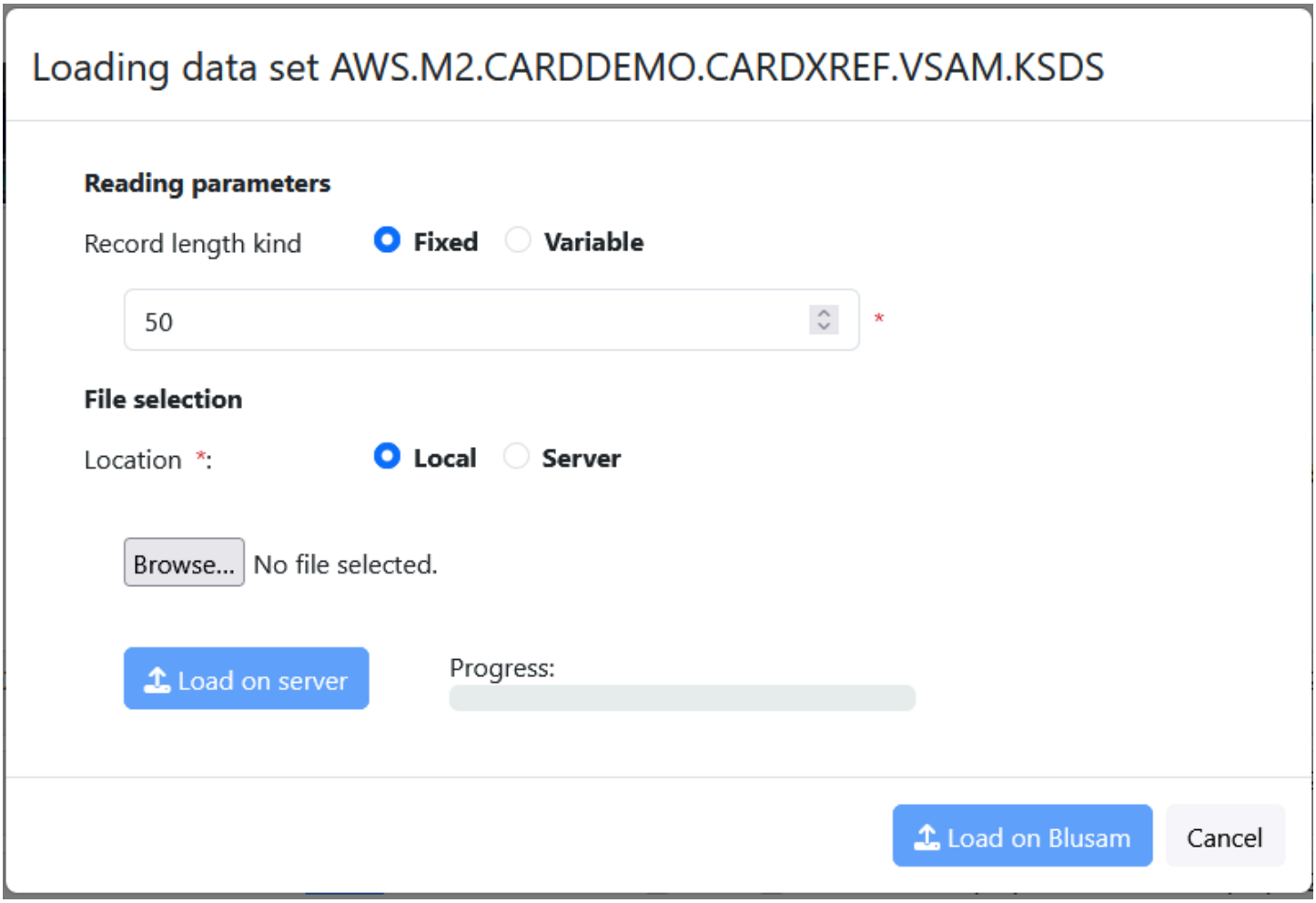


Caricamento dei record in un set di dati

Per caricare i record in un set di dati, scegli Azioni, quindi scegli Carica.



Viene visualizzata una finestra con le opzioni di caricamento.

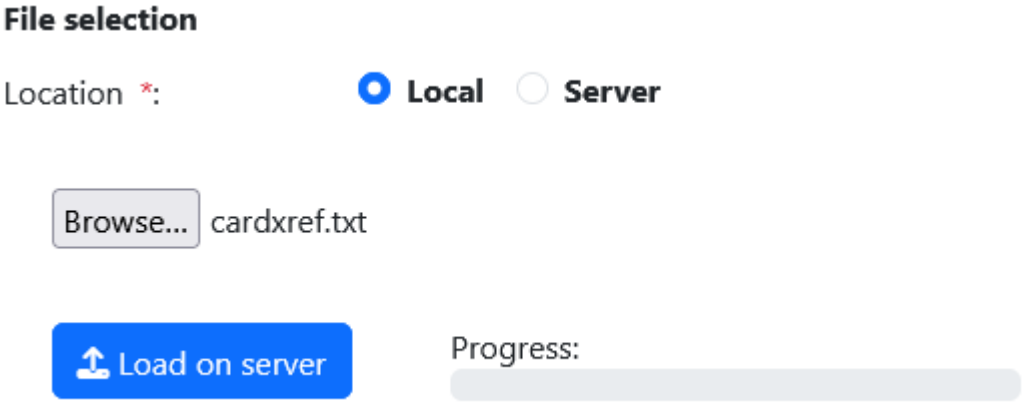


Inizialmente, entrambi i pulsanti Load on server e Load on Blusam sono disabilitati.

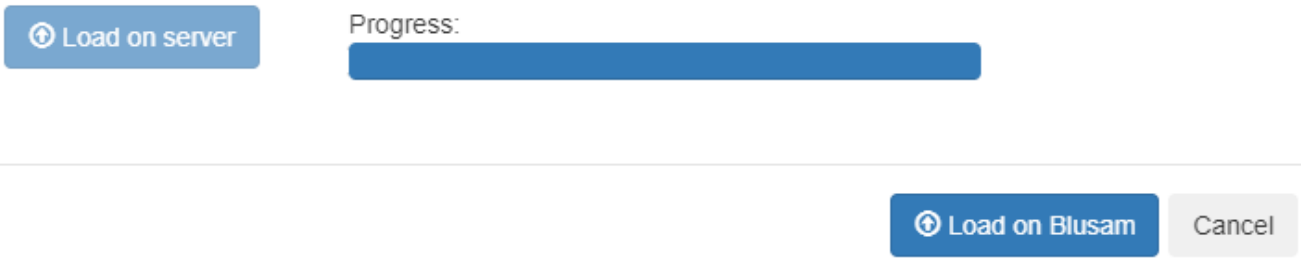
Parametri di lettura:

- Tipo di lunghezza del record:
  - Lunghezza di record fissa o variabile: utilizzate il pulsante di opzione per specificare se l'esportazione del set di dati precedente utilizza record a lunghezza fissa o record a lunghezza variabile (i record dovrebbero iniziare con byte RDW). Se scegliete Fisso, la lunghezza del record deve essere specificata (in byte) come valore intero positivo nel campo di immissione. Il valore deve essere precompilato con le informazioni provenienti dal set di dati. Se scegli Variabile, il campo di input specificato scompare.
  - Selezione del file:
    - Locale: scegliete il file del set di dati dal computer locale, utilizzando il selettore di file riportato di seguito (Nota: il selettore di file utilizza le impostazioni locali del browser per stampare i messaggi, qui in francese, ma dal vostro lato potrebbe apparire diverso, come previsto). Dopo

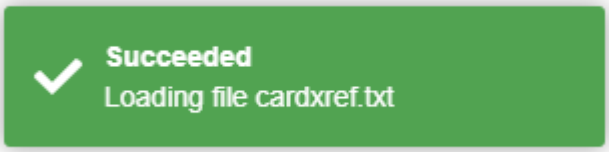
aver effettuato la selezione, la finestra viene aggiornata con il nome del file di dati e il pulsante Carica sul server è abilitato:



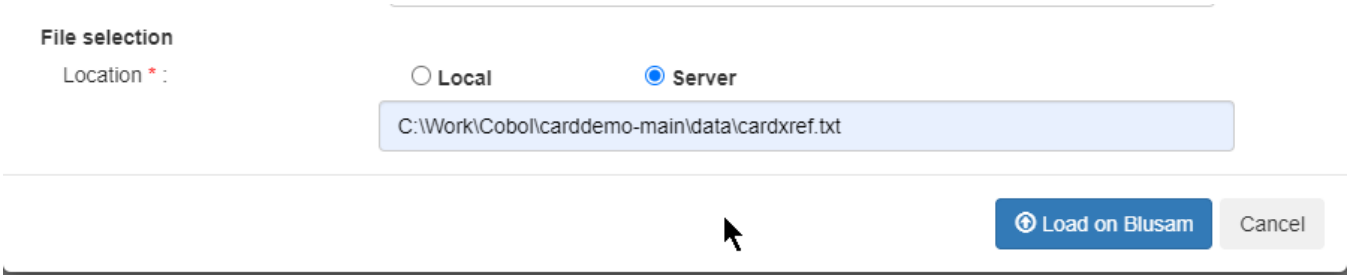
Scegliete Carica sul server. Dopo che la barra di avanzamento raggiunge la fine, il pulsante Load on Blusam viene abilitato:



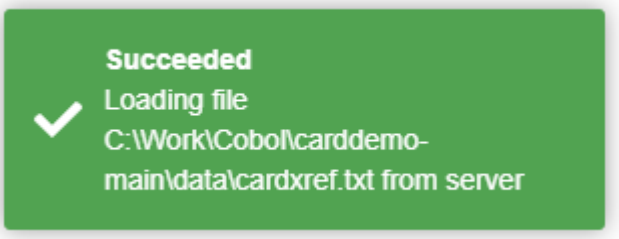
Per completare il processo di caricamento sullo Blusam storage, scegli Load on Blusam. Altrimenti, scegli Cancel (Annulla). Se scegli di continuare con il processo di caricamento, una notifica apparirà nell'angolo in basso a destra dopo il completamento del processo di caricamento:



- Server: selezionando questa opzione viene visualizzato un campo di immissione mentre il pulsante Carica sul server scompare. Nel campo di input è necessario specificare il percorso del file del set di dati sul Blusam server (ciò presuppone che sia stato prima trasferito il file specificato sul Blusam server). Dopo aver specificato il percorso, Load on Blusam viene abilitato:

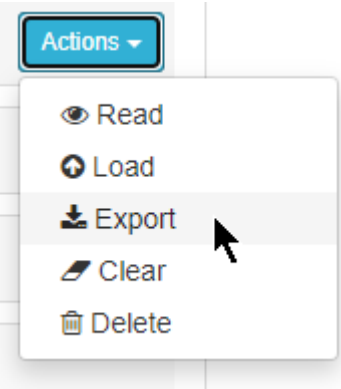


Per completare il processo di caricamento, scegliete Load on Blusam. Altrimenti, scegli Cancel (Annulla). Se scegli di procedere con il caricamento, al termine del processo di caricamento viene visualizzata una notifica. La notifica è diversa dal caricamento dal browser in quanto visualizza il percorso del server del file di dati seguito dalle parole dal server:

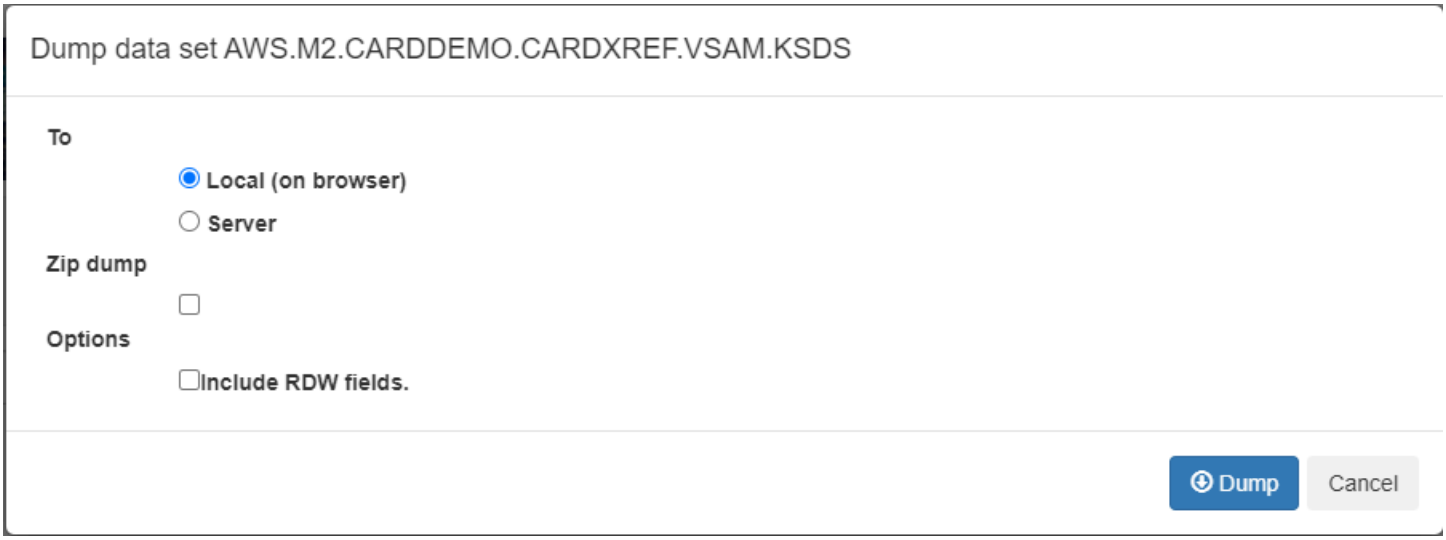


Esportazione di record da un set di dati

Per esportare i record del set di dati, scegli Azioni nella riga del set di dati corrente, quindi scegli Esporta:



Viene visualizzata la seguente finestra pop-up.



Opzioni:

A: a scelta del pulsante di opzione, per scegliere la destinazione di esportazione, da scaricare nel browser (locale (sul browser)) o in una determinata cartella sul Server che ospita l'applicazione BAC. Se scegli di esportare utilizzando l'opzione Server, verrà visualizzato un nuovo campo di immissione:



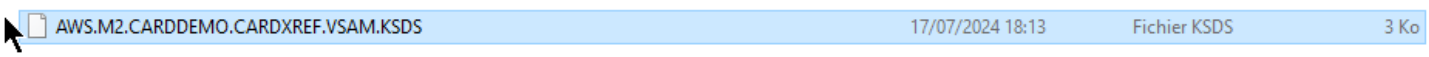
Come indica l'asterisco rosso a destra del campo di immissione, è obbligatorio fornire una posizione valida per la cartella sul server (il pulsante Dump sarà inattivo finché non è stata fornita alcuna posizione per la cartella).

Per esportare sul server, è necessario disporre dei diritti di accesso sufficienti per il file system del server, se si prevede di manipolare il file del set di dati esportato dopo l'esportazione.

Zip dump: una casella di controllo che produce un archivio compresso anziché un file raw.

Opzioni: per includere un Record Descriptor Word (RDW) all'inizio di ogni record nel set di dati esportato, nel caso di un set di dati di record a lunghezza variabile, scegli Includi campi RDW.

Per avviare il processo di esportazione del set di dati, scegliete Dump. Se scegli di esportare nel browser, controlla la cartella di download per il file del set di dati di esportazione. Il file avrà lo stesso nome del set di dati:

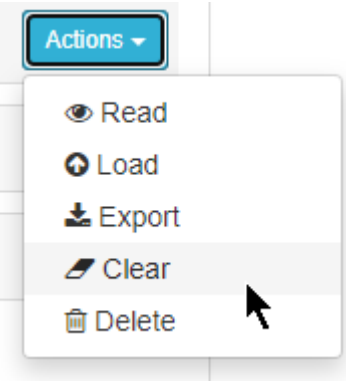


Nota:

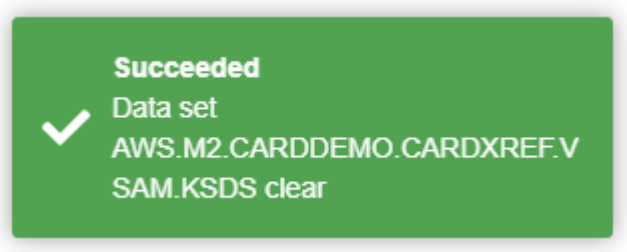
- Per KSDS, i record verranno esportati seguendo l'ordine delle chiavi primarie.
- Per ESDS e RRDS, i record verranno esportati secondo l'ordine RBA (Relative Byte Address).
- Per tutti i tipi di set di dati, i record verranno esportati come matrici binarie non elaborate (non verrà effettuata alcuna conversione), garantendo la compatibilità diretta con le piattaforme legacy.

Cancellazione dei record da un set di dati

Per cancellare tutti i record da un set di dati, scegli Azioni, quindi scegli Cancella:

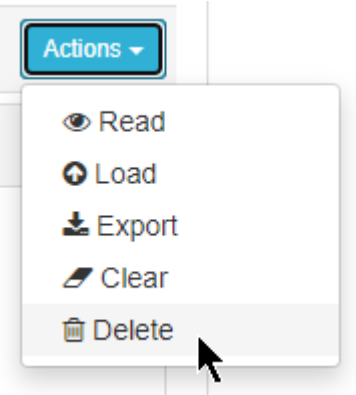


Dopo che tutti i record sono stati rimossi da un set di dati, viene visualizzata la seguente notifica.

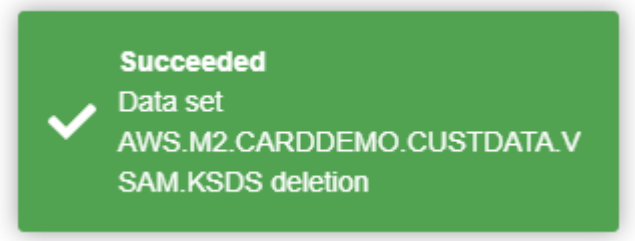


Eliminazione di un set di dati

Per eliminare un set di dati, scegli Azioni, quindi scegli Elimina:



Dopo aver eliminato un set di dati, viene visualizzata la seguente notifica:

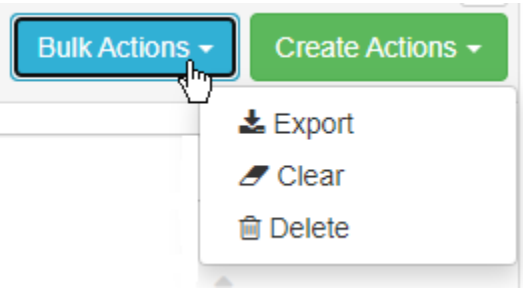


Operazioni in blocco

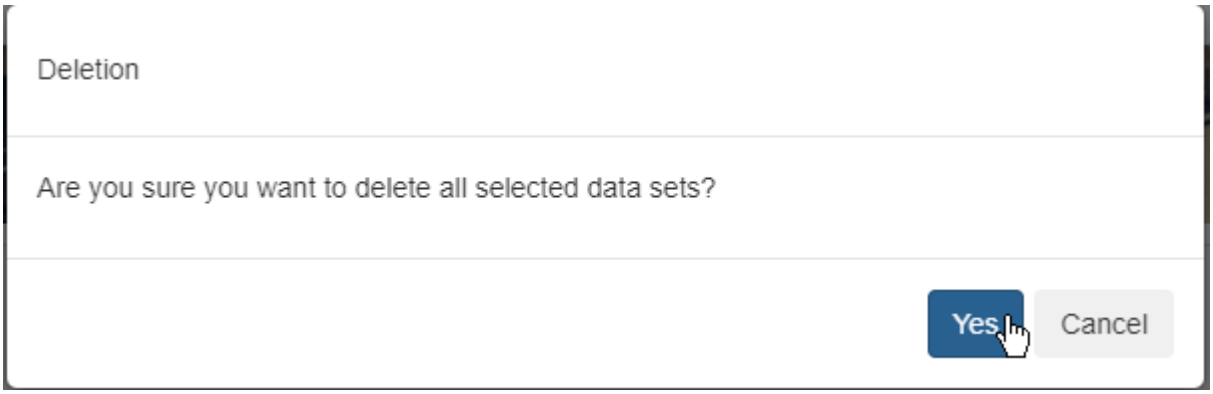
Sono disponibili tre operazioni in blocco sui set di dati:

- Esportazione
- Annulla
- Elimina

Le operazioni in blocco possono essere applicate solo a una selezione di set di dati (è necessario selezionare almeno un set di dati); la selezione dei set di dati viene effettuata spuntando le caselle di selezione a sinistra delle righe dei set di dati, nella tabella dell'elenco dei set di dati. La selezione di almeno un set di dati abiliterà l'elenco a discesa Azioni in blocco:



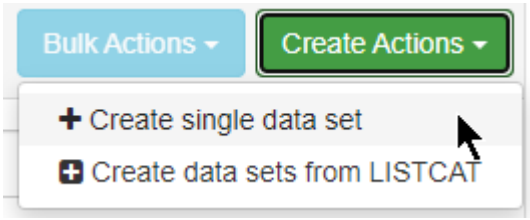
A parte il fatto che le azioni indicate si applicano a una selezione di set di dati anziché a uno solo, le azioni sono simili a quelle sopra descritte, quindi consulta la documentazione dedicata alle azioni per i dettagli. Il contenuto del testo delle finestre pop-up sarà leggermente diverso per riflettere la natura di massa. Ad esempio, quando si tenta di eliminare diversi set di dati, la finestra pop-up avrà il seguente aspetto:



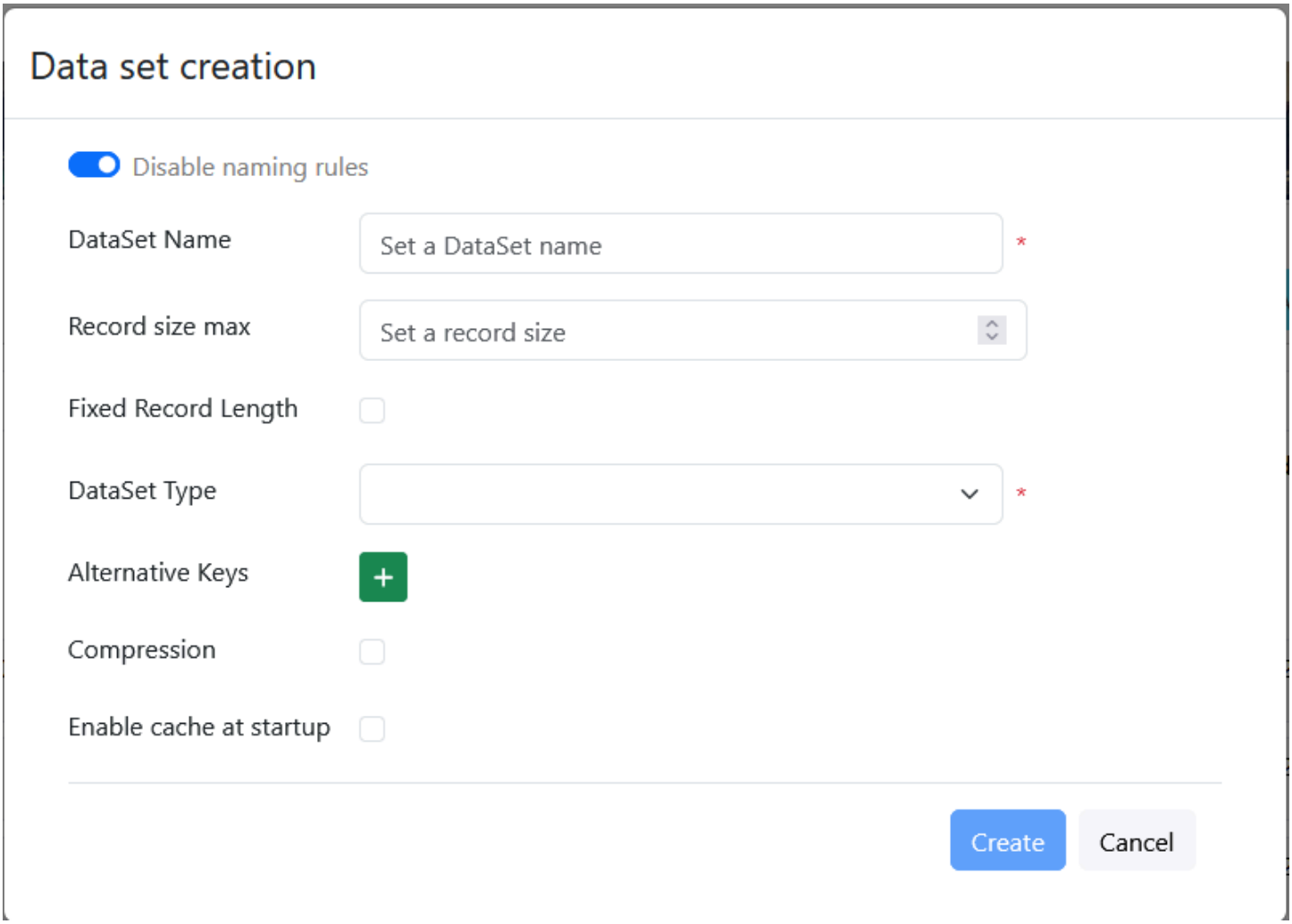
Creazione di operazioni

Crea un singolo set di dati

Scegli Azioni, quindi scegli Crea un unico set di dati:

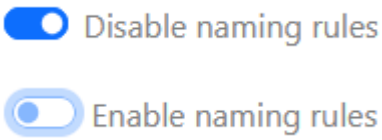


Il modulo di creazione del set di dati verrà quindi visualizzato come una finestra pop-up:

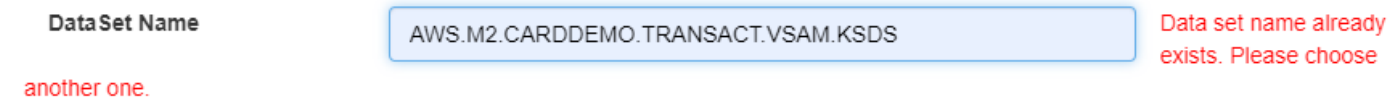


È possibile specificare i seguenti attributi per la definizione del set di dati:

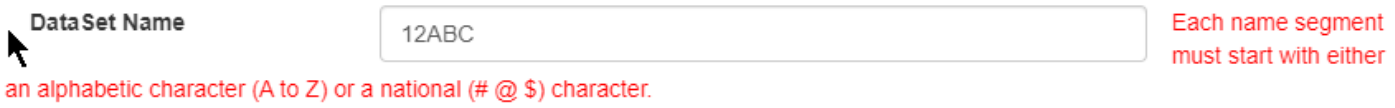
- Abilitazione e disabilitazione delle regole di denominazione: utilizza il widget «Disabilita regole di denominazione/Abilita regole di denominazione» per disabilitare e abilitare le convenzioni di denominazione dei set di dati. Ti consigliamo di lasciare l'interruttore sul valore predefinito, con le regole di denominazione dei set di dati abilitate (il widget Toggle dovrebbe mostrare «Disabilita le regole di denominazione»):

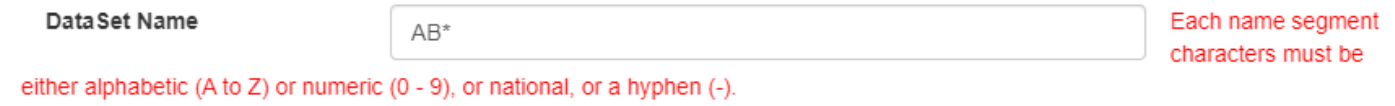
  

- Nome del set di dati: il nome del set di dati. Se si specifica un nome già in uso, viene visualizzato il seguente messaggio di errore.

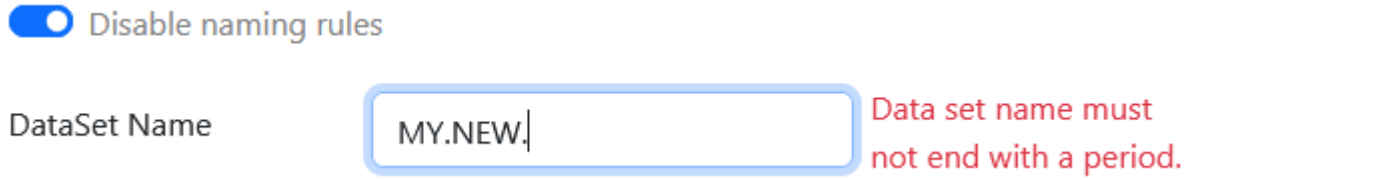


Il nome deve inoltre rispettare la convenzione di denominazione se è abilitato:





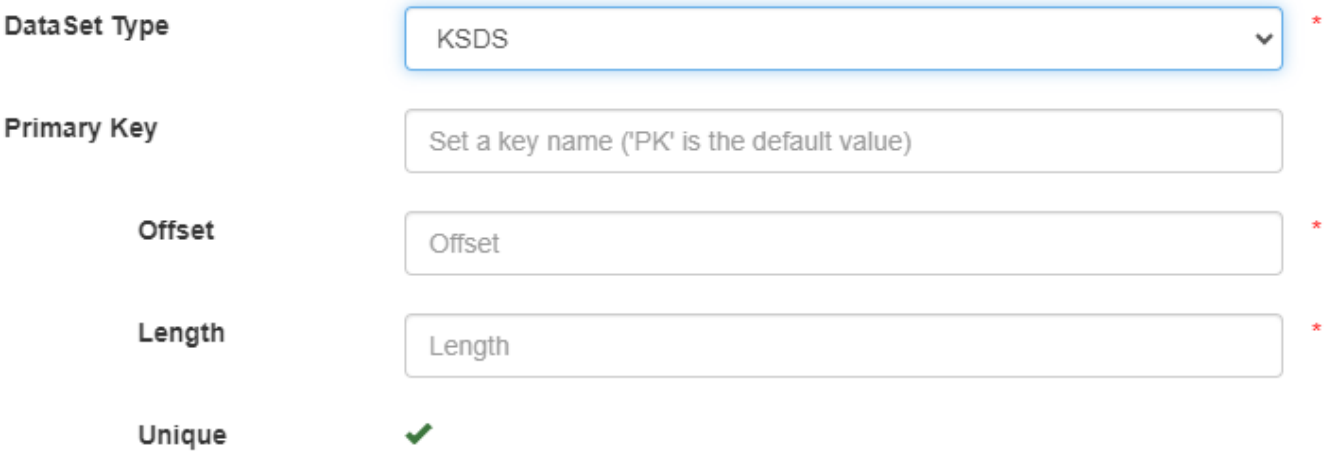




- Dimensione massima del record: deve essere un numero intero positivo che rappresenta la dimensione del record per un set di dati con record a lunghezza fissa. Puoi lasciarlo vuoto per i set di dati con record a lunghezza variabile.
- Record a lunghezza fissa: una casella di controllo per specificare se la lunghezza del record è fissa o variabile. Se selezionato, il set di dati conterrà record a lunghezza fissa, altrimenti la lunghezza dei record sarà variabile.

  Quando si importano dati preesistenti in un set di dati di record a lunghezza variabile, i record legacy forniti devono contenere il Record Descriptor Word (RDW) che indica la lunghezza di ogni record.
- Tipo di set di dati: un elenco a discesa per specificare il tipo di set di dati corrente. Sono supportati i seguenti tipi.
  - SDS
  - ESDS di grandi dimensioni
  - KDS

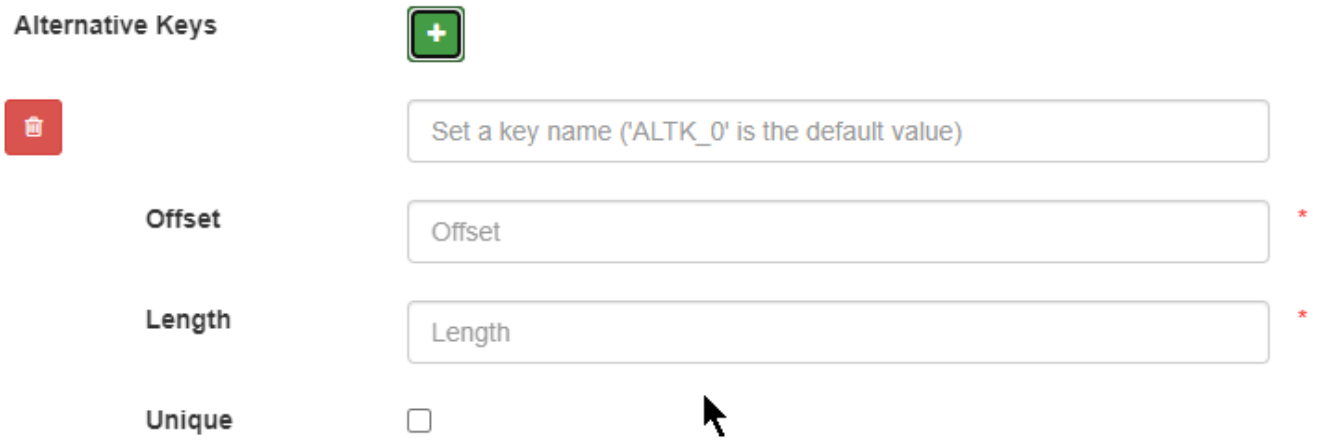
Per KSDS, devi specificare la chiave primaria:



Per la chiave primaria, specificare quanto segue:

- Nome: Questo campo è facoltativo. Il valore predefinito è **PK**.
- Offset: l'offset in base 0 della chiave primaria all'interno del record. L'offset deve essere un numero intero positivo. Questo campo è obbligatorio.
- Lunghezza: la lunghezza della chiave primaria. Questa lunghezza deve essere un numero intero positivo. Questo campo è obbligatorio.

Per KSDS ed ESDS, puoi facoltativamente definire una raccolta di chiavi alternative, scegliendo il pulsante Plus davanti all'etichetta Alternate Keys. Ogni volta che si sceglie quel pulsante, nel modulo di creazione del set di dati viene visualizzata una nuova sezione di definizione delle chiavi alternativa:



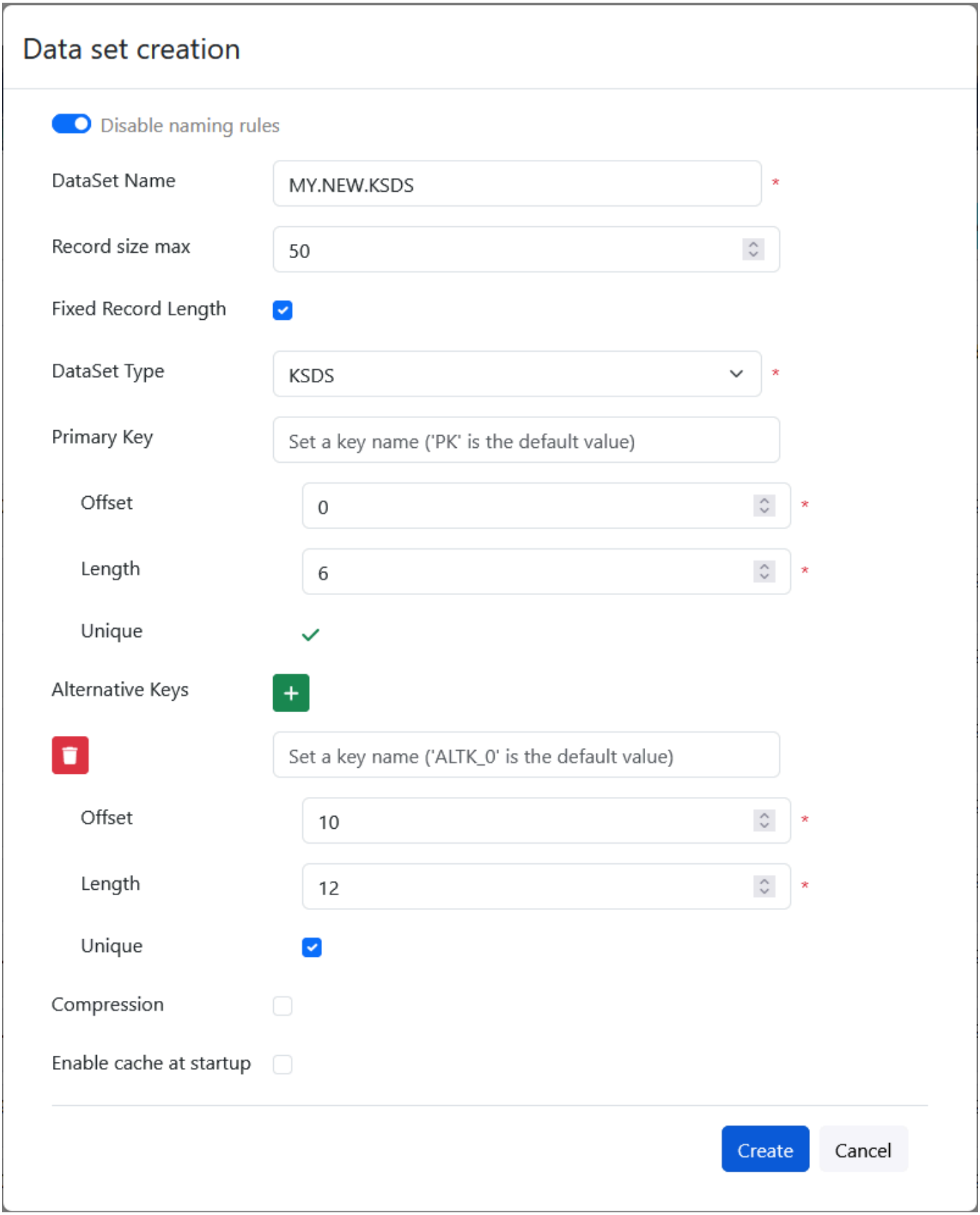
Per ogni chiave alternativa, devi fornire:

- Nome: Questo campo è facoltativo. Il valore predefinito è**`ALTK_#`**, dove # rappresenta un contatore ad incremento automatico che parte da 0.
- Offset: l'offset in base 0 della chiave alternativa all'interno del record. Deve essere un numero intero positivo. Questo campo è obbligatorio.
- Lunghezza: la lunghezza della chiave alternativa. Questa lunghezza deve essere un numero intero positivo. Questo campo è obbligatorio.
- Unico: una casella di controllo per indicare se la chiave alternativa accetterà voci duplicate. Se selezionata, la chiave alternativa verrà definita come unica (NON accetta immissioni chiave duplicate). Questo campo è obbligatorio.

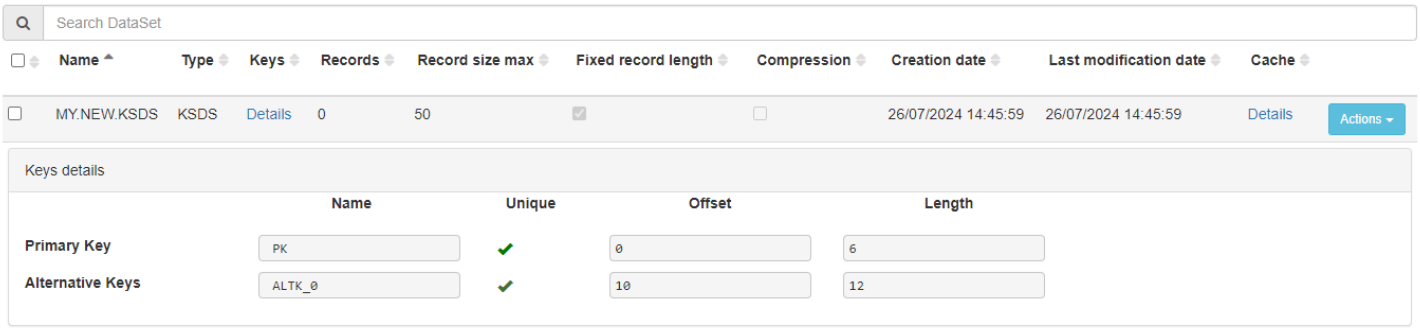
Per rimuovere la definizione di chiave alternativa, usa il pulsante del cestino a sinistra.

- Compressione: una casella di controllo per specificare se la compressione verrà utilizzata per memorizzare il set di dati.
- Abilita cache all'avvio: una casella di controllo per specificare se il set di dati deve essere caricato nella cache all'avvio dell'applicazione.

Dopo aver specificato le definizioni degli attributi, scegliete Crea per procedere:


Data set creation
Disable naming rules
DataSet Name
MY.NEW.KSDS
Record size max
50
Fixed Record Length
DataSet Type
KSDS
Primary Key
Set a key name ('PK' is the default value)
Offset
0
Length
6
Unique
Alternative Keys
Set a key name ('ALTK_0' is the default value)
Offset
10
Length
12
Unique
Compression
Enable cache at startup
Create
Cancel

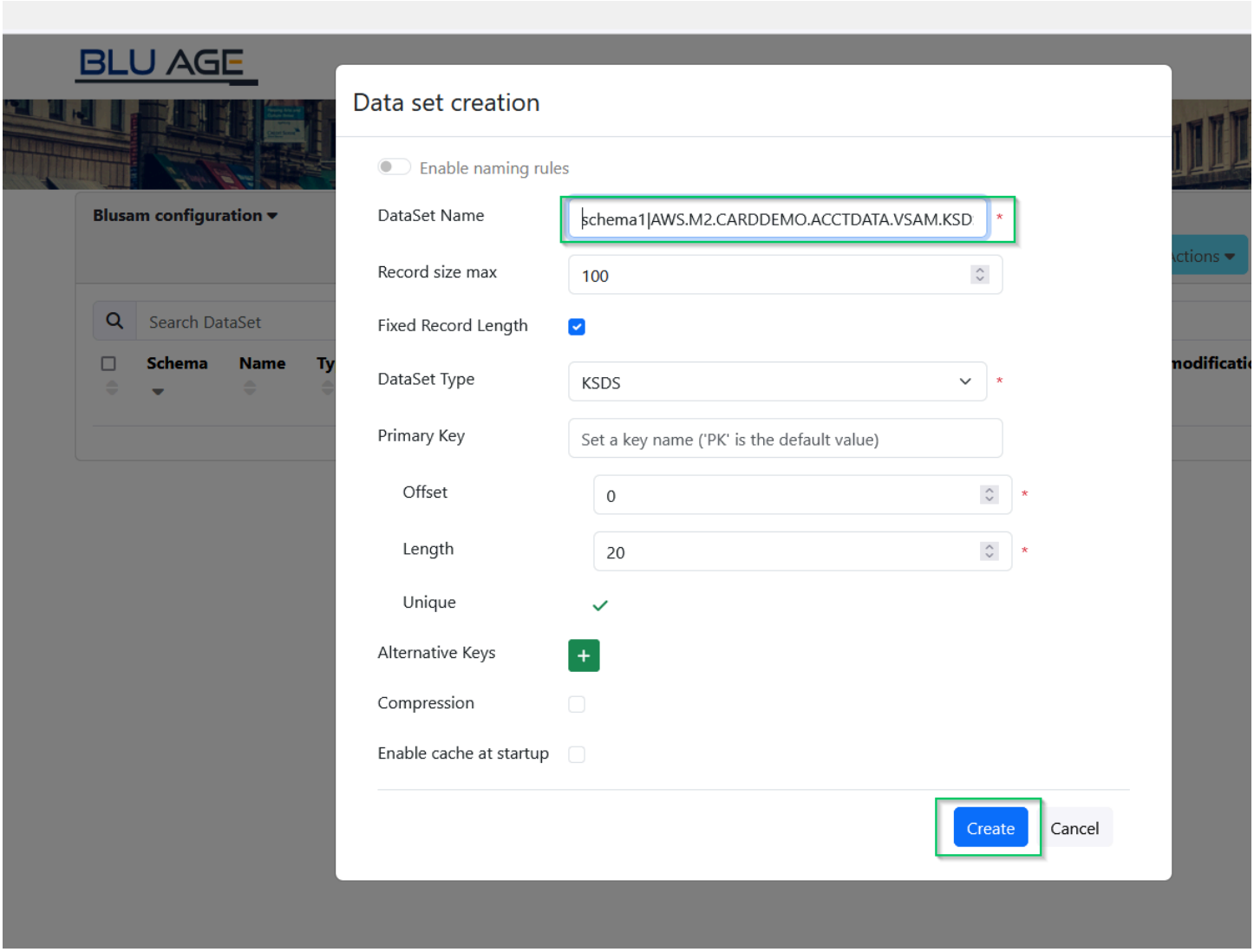
La finestra di creazione verrà chiusa e verrà visualizzata la home page che mostra l'elenco dei set di dati. È possibile visualizzare i dettagli del set di dati appena creato.



Crea un singolo set di dati in modalità multischema

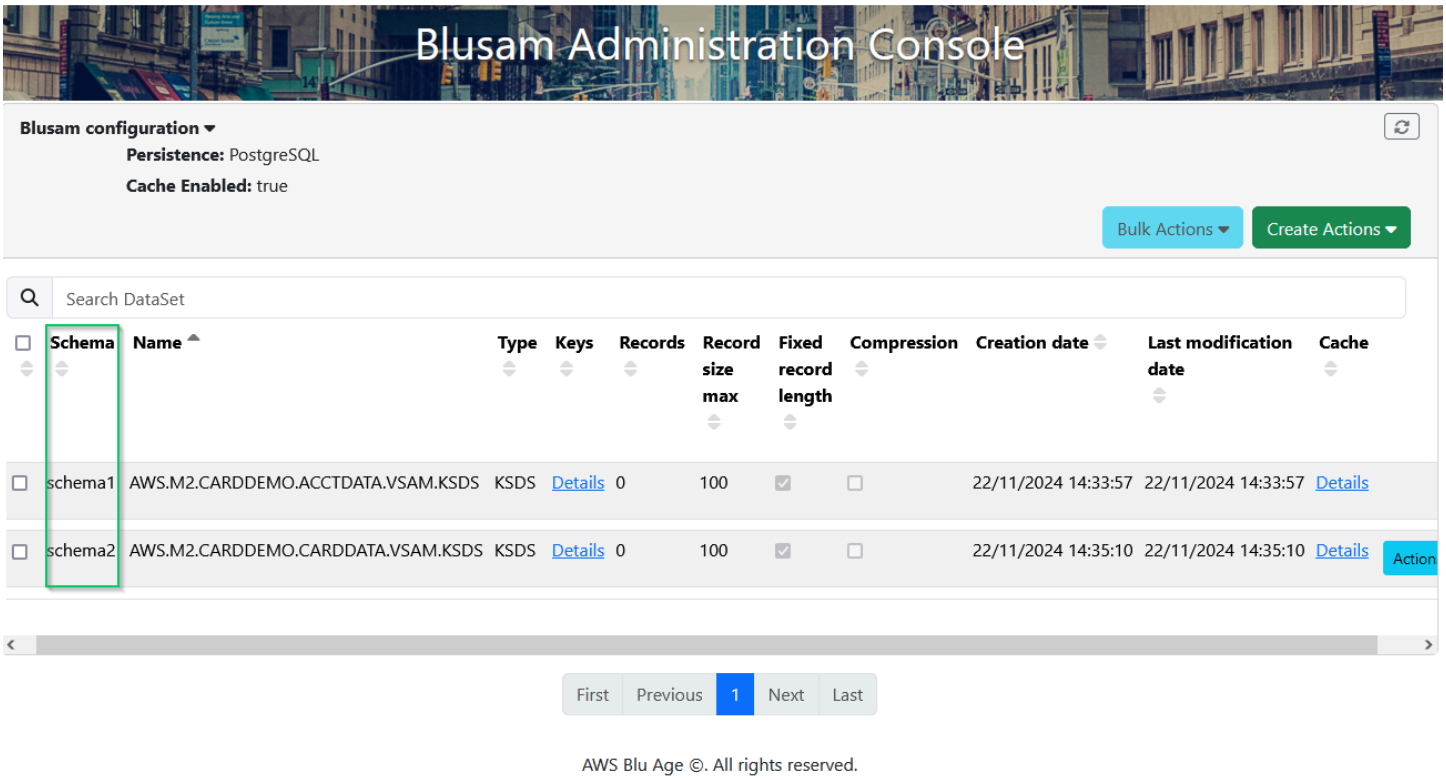
Un set di dati può essere creato in modalità multischema anteponendo al nome del set di dati il nome dello schema seguito da un simbolo pipe (|) (ad esempio,). `schema1|AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS`

> **Note**
>
> Lo schema utilizzato per creare il set di dati deve essere specificato nella `application-main.yml` configurazione. Per ulteriori informazioni, consulta the section called "Proprietà di configurazione multischema".



Se non viene fornito alcun prefisso dello schema, il set di dati verrà creato nello schema predefinito specificato nell'URL dell'Blusamorigine dati nella configurazione Datasource. Blusam Se non è specificato alcuno schema nell'URL dell'Blusamorigine dati, per impostazione predefinita viene utilizzato lo schema «pubblico».

> **Note**
>
> In modalità multischema, la console BAC visualizza le informazioni sullo schema del set di dati nella prima colonna.



Crea set di dati da LISTCAT

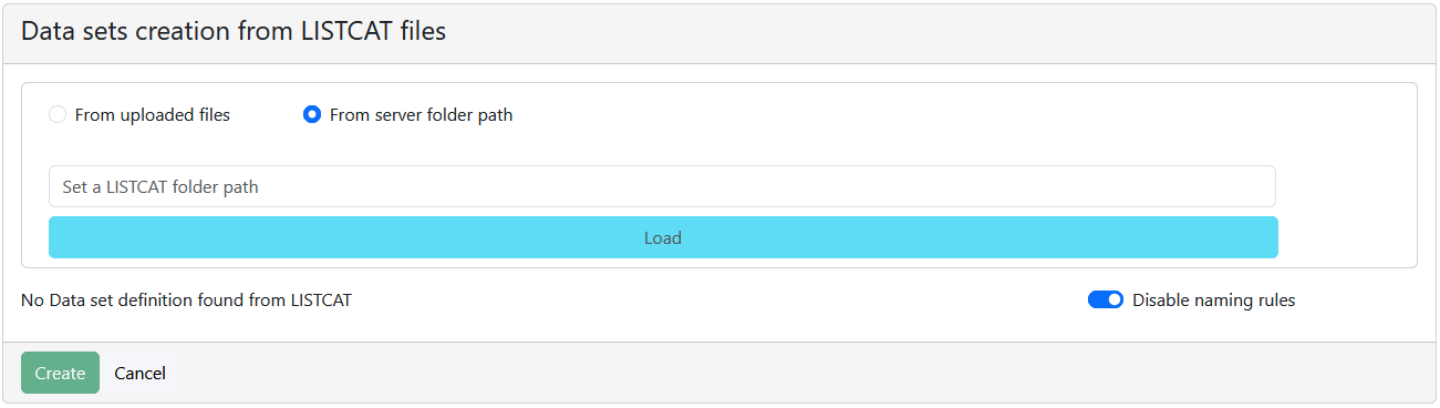
Questa funzionalità consente di sfruttare i file JSON LISTCAT creati durante il processo di trasformazione di AWS Transform for mainframe utilizzando AWS Transform for mainframe refactor Transformation Center come risultato dell'analisi dell'esportazione LISTCAT dalle piattaforme legacy: le esportazioni LISTCAT vengono analizzate e trasformate in file JSON che contengono le definizioni dei set di dati (nomi, tipo di set di dati, definizioni delle chiavi e se la lunghezza del record è fissa o variabile).

Avere i file JSON LISTCAT consente di creare set di dati direttamente senza dover inserire manualmente tutte le informazioni richieste per i set di dati. Puoi anche creare direttamente una raccolta di set di dati invece di doverli creare uno per uno.

Se non è disponibile alcun file LISTCAT JON per il progetto (ad esempio, perché nessun file di esportazione LISTCAT era disponibile al momento della trasformazione), puoi sempre crearne uno manualmente, a condizione di rispettare il formato JSON LISTCAT dettagliato nell'appendice.

Dall'elenco a discesa Crea azioni, scegli Crea set di dati da LISTCAT.

Verrà visualizzata la seguente pagina dedicata:



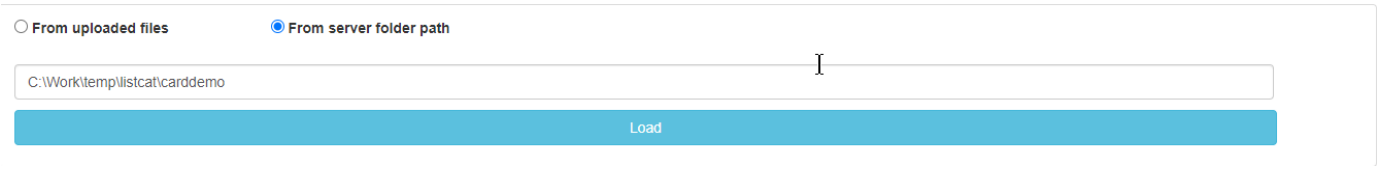
In questa fase, il pulsante Carica è disabilitato, come previsto.

Utilizzate i pulsanti di opzione per specificare come desiderate fornire i file JSON LISTCAT. Sono disponibili due opzioni:

- Puoi usare il tuo browser per caricare i file JSON.
- È possibile selezionare i file JSON da una cartella sul server. Per scegliere questa opzione, devi prima copiare i file JSON nel percorso della cartella specificato sul server con i diritti di accesso adeguati.

Per utilizzare i file JSON sul server

1. Imposta il percorso della cartella sul server, puntando alla cartella contenente i file JSON LISTCAT:



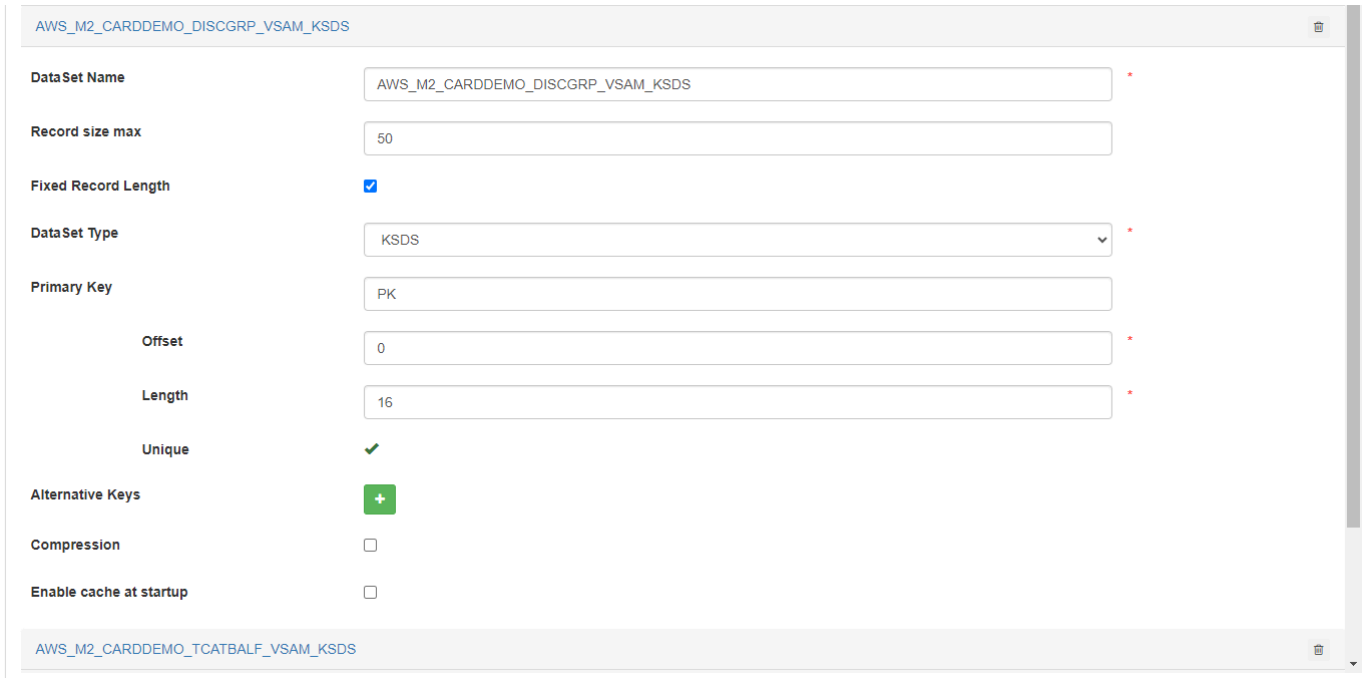
2. Scegli il pulsante Carica. Tutte le definizioni dei set di dati riconosciute verranno elencate in una tabella:



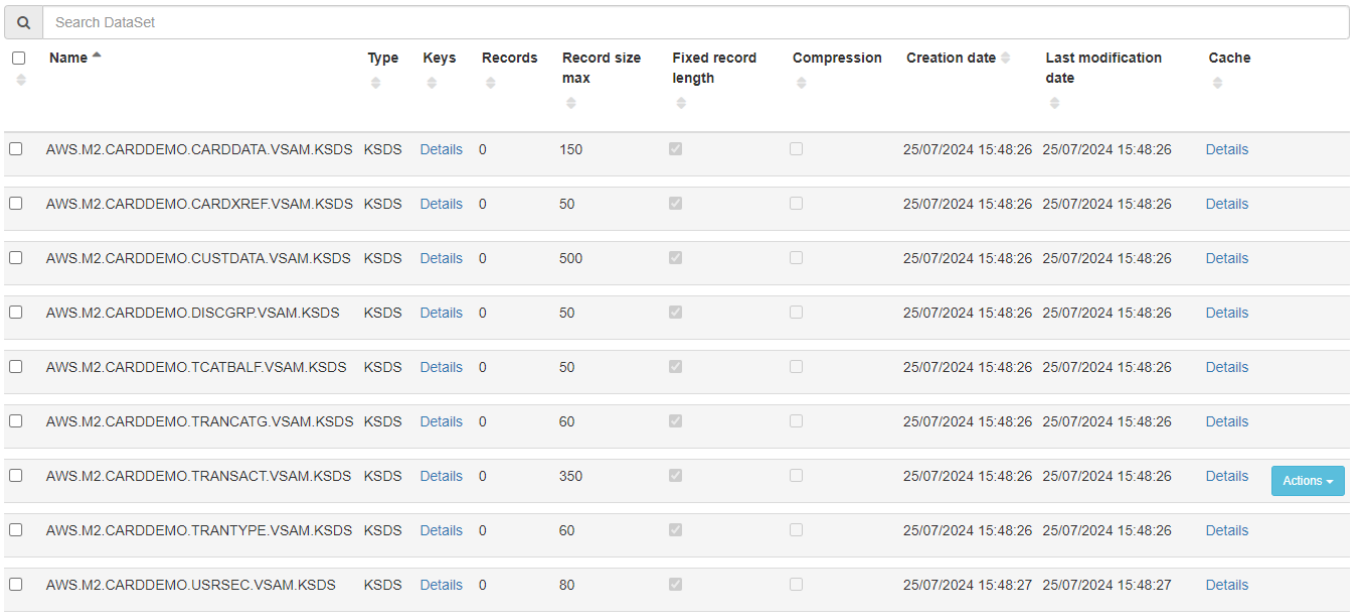
Ogni riga rappresenta una definizione di set di dati. È possibile utilizzare il pulsante del cestino per rimuovere una definizione di set di dati dall'elenco.

> **Important**
>
> La rimozione dall'elenco è immediata, senza alcun messaggio di avviso.

3. Il nome a sinistra è un link. Puoi sceglierlo per mostrare o nascondere i dettagli della definizione del set di dati, che è modificabile. Puoi modificare liberamente la definizione, partendo dalla base del file JSON analizzato.

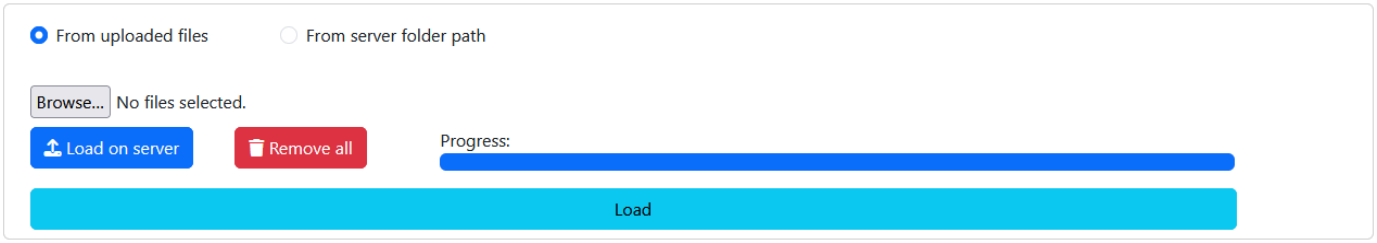


4. Per creare tutti i set di dati, scegli Crea. Tutti i set di dati verranno creati e visualizzati nella pagina dei risultati dei set di dati. I set di dati appena creati avranno tutti 0 record.



Per caricare file sul server

1. Questa opzione è simile all'utilizzo dei file dal percorso della cartella del server, ma in questo caso è necessario prima caricare i file utilizzando il selettore di file. Seleziona tutti i file da caricare dal computer locale, quindi scegli Carica sul server.



2. Quando la barra di avanzamento raggiunge la fine, tutti i file sono stati caricati correttamente sul server e il pulsante Carica è abilitato. Scegliete il pulsante Carica e utilizzate le definizioni dei set di dati rilevati come spiegato in precedenza.

Formato JSON LISTCAT

Il formato JSON LISTCAT è definito dai seguenti attributi:

- opzionale «catalogiD»: identificatore del catalogo precedente come String o «default» per il catalogo predefinito.
- «identifier»: il nome del set di dati, come stringa.
- «isIndexed»: un flag booleano per indicare KSDS: true per KSDS, false altrimenti.
- «isLinear»: un flag booleano per indicare ESDS: true per ESDS, false altrimenti.
- «isRelative»: un flag booleano per indicare RRDS: true per RRDS, false altrimenti
- Nota: «isIndexed», «isLinear» e «isRelative» si escludono a vicenda.
- "isFixedLengthRecord»: un flag booleano: impostato su true se la lunghezza fissa registra un set di dati, false in caso contrario.
- "avgRecordSize«: Dimensione media dei record in byte, espressa come numero intero positivo.
- "maxRecordSize«: Dimensione massima del record in byte, espressa come numero intero. Deve essere uguale alla dimensione del record avgRecordSize a lunghezza fissa.
- solo per KSDS: definizione obbligatoria della chiave primaria (come oggetto annidato)
  - etichettato «PrimaryKey»
  - «offset»: offset in base a 0 byte per la chiave primaria del record.
  - «length»: lunghezza in byte della chiave primaria.
  - «unique»: deve essere impostato su true per la chiave primaria.
- per KSDS/ESDS, raccolta di chiavi alternative (come raccolta di oggetti annidati):
  - etichettato «AlternateKeys»
  - Per ogni chiave alternativa:
    - «offset»: offset in base a 0 byte per la chiave alternativa nel record.
    - «length»: lunghezza in byte della chiave alternativa.
    - «unique»: deve essere impostato su true per la chiave alternativa, se la chiave non accetta voci duplicate, false in caso contrario.
- se non sono presenti chiavi alternative, fornisci una raccolta vuota:

```
alternateKeys: []
```

Di seguito è riportato un esempio di file JSON KSDS LISTCAT.

```
{
  "catalogId": "default",
```

```
  "identifier": "AWS_M2_CARDDEMO_CARDXREF_VSAM_KSDS",
  "isIndexed": true,
  "isLinear": false,
  "isRelative": false,
  "isFixedLengthRecord": true,
  "avgRecordSize": 50,
  "maxRecordSize": 50,
  "primaryKey": {
    "offset": 0,
    "length": 16,
    "unique": true
  },
  "alternateKeys": [
    {
      "offset": 25,
      "length": 11,
      "unique": false
    }
  ]
}
```

Configurare la configurazione per AWS Transform for mainframe Runtime

AWS Transform for mainframe Runtime 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 AWS Transform for mainframe Runtime 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
- AWS I segreti di Transform for mainframe Runtime
- Altri file (groovy, sql, ecc.)

- Applicazione web aggiuntiva
- Abilita le proprietà per AWS Transform for mainframe Runtime
- Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
- Configurare la sicurezza 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 : pgsql #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-[redacted].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:[redacted]

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. [Abilita le proprietà per AWS Transform for mainframe Runtime]

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.

AWS I segreti di Transform for mainframe Runtime

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 di inserire un riferimento a tale segreto nella configurazione YAML, in modo che il contenuto segreto venga raccolto all'avvio di Apache Tomcat.

Segreti per Aurora

La configurazione del database Aurora (per JICSBlusam, customer db e così via) 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.

Riferimenti YAML ai segreti

`application-main.yml`Possono fare riferimento all'ARN segreto per varie risorse:

Database JICS

Credenziali del database JICS con `spring.aws.jics.db.secret`

```
spring:
   aws:
     jics:
       db:
         dbname: jics
         secret: arn:aws:secretsmanager:XXXX
```

Chiavi segrete del database JICS supportate:

Chiave privata	Descrizione della chiave segreta
host	Il nome dell'host
port	La porta
dbname	Il nome del database
username	Il nome utente
password	La password
engine	Motore di database: Postgres, Oracle, Db2, Microsoft SQL Server
Schema attuale	Schema specifico da utilizzare (solo supporto Db2)
Connessione SSL	Se utilizzare la connessione SSL (solo supporto Db2)
sslTrustStoreLocation	La posizione del truststore sul client (solo supporto Db2)
sslTrustStorePassword	La password per il truststore sul client (solo supporto Db2)

> **Note**
>
> Il nome del database viene fornito nel segreto o nel riferimento yaml. `spring.aws.jics.db.dbname`

Database Blusam

Blusamcredenziali del database con `spring.aws.client.bluesam.db.secret`

```
spring:
   aws:
```

```
    client:
      bluesam:
        db:
          dbname: bluesam
          secret: arn:aws:secretsmanager:XXXX
```

Chiavi segrete Blusam del database supportate:

Chiave privata	Descrizione della chiave segreta
host	Il nome dell'host
port	La porta
dbname	Il nome del database
username	Il nome utente
password	La password
engine	Motore di database: Postgres

> **Note**
>
> Il nome del database viene fornito nel segreto o nel riferimento yaml. `spring.aws.client.bluesam.db.dbname`

Database client

Il client `application-profile.yml` può fare riferimento all'ARN segreto per il database del client. Ciò richiede una proprietà aggiuntiva per elencare i nomi delle sorgenti dati. `spring.aws.client.datasources.names` Per ogni nome di origine dati, `ds_name` specificare l'ARN segreto nella seguente proprietà:. `spring.aws.client.datasources.ds_name.secret` Esempio:

```
spring:
   aws:
     client:
       datasources:
```

```
        names: primary,host
        primary:
          secret: arn:aws:secretsmanager:XXXX
        host:
          dbname: hostdb
          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: hostdb:

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.

Chiavi segrete del database client supportate:

Chiave privata	Descrizione della chiave segreta
host	Il nome dell'host
port	La porta
dbname	Il nome del database
username	Il nome utente
password	La password
engine	Motore di database: Postgres, Oracle, Db2, Microsoft SQL Server
Schema attuale	Schema specifico da utilizzare (solo supporto Db2)
Connessione SSL	Se utilizzare la connessione SSL (solo supporto Db2)
sslTrustStoreLocation	La posizione del truststore sul client (solo supporto Db2)

Chiave privata	Descrizione della chiave segreta
sslTrustStorePassword	La password per il truststore sul client (solo supporto Db2)

Database di utilità PGM

`application-utility-pgm.yml`Possono fare riferimento all'ARN segreto per varie risorse.

- `spring.aws.client.datasources.primary`
  - `secret`

    ARN segreto per il database dell'applicazione.

Tipo: stringa

- `type`

Nome completo dell'implementazione del pool di connessioni da utilizzare.

Tipo: stringa

Impostazione predefinita: `com.zaxxer.hikari.HikariDataSource`

- `spring.aws.client.utility.pgm.datasources`
  - `names`

Elenco dei nomi delle fonti di dati.

Tipo: stringa

- `dsname`
  - `dbname`

Nome dell'host.

Tipo: stringa

- `secret`

ARN segreto del database host.

Tipo: stringa

- `type`

Nome completo dell'implementazione del pool di connessioni da utilizzare.

Tipo: stringa

Impostazione predefinita: `com.zaxxer.hikari.HikariDataSource`

Per un segreto basato su più origini dati:

```
spring:
   aws:
     client:
       primary:
         secret: arn:aws:secretsmanager:XXXX
         type: dataSourceType
       utility:
         pgm:
           datasources:
             names: dsname1,dsname2,dsname3
               dsname1:
                 dbname: dbname1
                 secret: arn:aws:secretsmanager:XXXX
                 type: dataSourceType
               dsname2:
                 dbname: dbname2
                 secret: arn:aws:secretsmanager:XXXX
                 type: dataSourceType
               dsname3:
                 dbname: dbname3
                 secret: arn:aws:secretsmanager:XXXX
                 type: dataSourceType
```

Nessuna chiave segreta supportata da XA

- motore () postgres/oracle/db2/mssql

- port
- dbname
- Schema corrente
- username
- password
- url
- Connessione SSL
- sslTrustStoreLocation
- sslTrustStorePassword

Perché `postgres` solo il valore della chiave `sslMode` segreta (`disable/allow/prefer/require/verify-ca/verify-full`) e la proprietà `spring.aws.rds.ssl.cert-path` YAML consentono la connessione con SSL.

XA supportava le chiavi segrete

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)
- sslTrustStoreLocation
- sslTrustStorePassword

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

Il valore segreto ha la precedenza sulle proprietà YAML.

Super Admin BAC e JAC predefiniti

Puoi anche configurare application-main.yml per recuperare il nome utente e la password dell'utente super amministratore predefinito nel segreto di AWS Secrets Manager specificando l'ARN. L'esempio seguente mostra come dichiarare questo segreto in un file YAML.

```
spring:
   aws:
     client:
       defaultSuperAdmin:
         secret: arn:aws:secretsmanager:XXXX
```

Chiavi segrete del database super admin predefinite supportate:

Chiave privata	Descrizione della chiave segreta
username	Il nome utente.
password	La password.

OAuth2

Puoi anche configurare `application-main.yml` per recuperare il OAuth2 segreto del client specificando il provider e l'ARN. Gestione dei segreti AWS Il valore predefinito per la proprietà del provider è Amazon Cognito. Di seguito è riportato un esempio di configurazione per il OAuth2 provider Keycloak:

```
spring:
   aws:
     client:
       provider: keycloak
       keycloak:
         secret: arn:aws:secretsmanager:XXXX
```

In questo esempio, il client-secret per il OAuth2 provider Keycloak viene recuperato dall'ARN specificato in AWS Secrets Manager. Questa configurazione supporta più provider risolvendo dinamicamente il nome del provider e l'ARN segreto corrispondente.

Chiavi segrete supportate OAuth2 :

Chiave privata	Descrizione della chiave segreta
segreto del cliente	Il segreto generato dal server di autorizzazione durante il processo di registrazione dell'appl icazione.

Gestore segreto per le cache Redis

Il `application-main.yml` file può fare riferimento all'ARN segreto per le cache Redis. Quelle supportate sono:

- Credenziali Gapwalk Redis con `spring.aws.client.gapwalk.redis.secret`
- Credenziali Bluesam Redis con `spring.aws.client.bluesam.redis.secret`
- Bluesam blocca le credenziali Redis con `spring.aws.client.bluesam.locks.redis.secret`
- Dataset cataloga le credenziali Redis con `spring.aws.client.dataset.catalog.redis.secret`
- Credenziali ICS Redis con `spring.aws.client.jics.redis.secret`
- Credenziali Redis della sessione con `spring.aws.client.jics.redis.secret`
- Credenziali Redis del tracker di sessione con `spring.aws.client.session.tracker.redis.secret`
- JICS TS mette in coda le credenziali Redis con `spring.aws.client.jics.queues.ts.redis.secret`
- Credenziali Redis del checkpoint JCL con `spring.aws.client.jcl.checkpoint.redis.secret`
- I file Gapwalk bloccano le credenziali Redis con `spring.aws.client.gapwalk.files.locks.redis.secret`
- Blu4iv blocca le credenziali Redis con `spring.aws.client.blu4iv.locks.redis.secret`

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

```
spring:
```

```
  aws:
    client:
      gapwalk:
        redis:
          secret: arn:aws:secretsmanager:XXXX
      bluesam:
        locks:
          redis:
            secret: arn:aws:secretsmanager:XXXX
        redis:
          secret: arn:aws:secretsmanager:XXXX
      dataset:
        catalog:
          redis:
            secret: arn:aws:secretsmanager:XXXX
      jics:
        redis:
          secret: arn:aws:secretsmanager:XXXX
      session:
        tracker:
          redis:
            secret: arn:aws:secretsmanager:XXXX
      jics:
        queues:
          ts:
            redis:
              secret: arn:aws:secretsmanager:XXXX
      jcl:
        checkpoint:
          redis:
            secret: arn:aws:secretsmanager:XXXX
      gapwalk:
        files:
          locks:
            redis:
              secret: arn:aws:secretsmanager:XXXX
      blu4iv:
        locks:
          redis:
            secret: arn:aws:secretsmanager:XXXX
```

Chiavi segrete Redis supportate:

Chiave privata	Descrizione della chiave segreta
hostname	Il nome host del server Redis.
port	La porta del server Redis.
username	Il nome utente.
password	La password.

Gestore segreto per le impostazioni delle password SSL

Il `application-main.yml` file può fare riferimento all'ARN segreto per le impostazioni della password SSL. È supportato quanto segue.

- Credenziali SSL Gapwalk con `spring.aws.client.ssl.secret`

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

```
spring:
   aws:
     client:
       ssl:
         secret: arn:aws:secretsmanager:XXXX
```

Chiave privata	Descrizione della chiave segreta
trustStorePassword	La password del truststore.
keyStorePassword	La password del keystore.

Gestore segreto per le impostazioni delle password di IBM MQ

Il `application-main.yml` file può fare riferimento all'ARN segreto per le impostazioni IBM MQ. È supportato quanto segue.

- Le connessioni IBM MQ sono definite come un elenco, così come le credenziali:

`mq.queues.jmsMQQueueManagers[N].secret:`

N inizia da 0 per la prima connessione.

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

```
mq.queues.jmsMQQueueManagers[0].secret: Secret-0-ARN
mq.queues.jmsMQQueueManagers[1].secret: Secret-1-ARN
```

Per informazioni sul segreto ARNs, vedi Cosa c'è in un segreto di Secrets Manager?

Le proprietà definite nel segreto sovrascriveranno i valori corrispondenti nella configurazione `jmsMQ` YAML.

Se `queueManager` è impostato nel segreto, sovrascriverà il `mq.queues.jmsMQQueueManagers[N].jmsMQQueueManager` valore nel file YAML.

Chiave privata	Descrizione della chiave segreta
QueueManager	Il nome del gestore di code IBM MQ.
appName	Il nome dell'applicazione IBM MQ.
channel	Il nome del canale IBM MQ.
host	Il nome host IBM MQ.
port	La porta IBM MQ.
userId	Il nome utente IBM MQ.
password	La password utente IBM MQ.
maxPoolSize	La dimensione massima del pool IBM MQ.
sslCipherKey	La suite di crittografia SSL IBM MQ.

Database HDB

Il `application-jhdb.yml` file client può fare riferimento al segreto ARNs per i database di metadati JHDB. Ogni connessione al database richiede un nome univoco e un ARN segreto corrispondente contenente le credenziali di connessione. I nomi dei database sono definiti in un elenco separato da virgole, con un segreto individuale ARNs mappato a ciascun nome di database.

```
spring:
   aws:
     jhdb:
       cnxs:
         datasources:
           names: DBD1,DBD2
           DBD1:
             secret: arn:aws:secretsmanager:XXXX
           DBD2:
             secret: arn:aws:secretsmanager:XXXX
```

Chiavi segrete del database client supportate:

Chiave privata	Descrizione della chiave segreta
host	Il nome host.
port	La porta.
dbname	Nome del database.
username	Il nome utente.
password	La password.
engine	Motore di database: Postgres (ora supporta solo Postgres).
Schema attuale	Schema specifico da utilizzare.

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 YAML 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

AWS Transform for mainframe Runtime 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.

Alcune configurazioni aggiuntive specifiche del progetto possono anche essere aggiunte 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.yaml` come configurazione.
- `gapwalk-cl-command.war`: contiene il supporto per i programmi di utilità AS/400 e li utilizza `application-cl-command.yaml` come configurazione.
- `gapwalk-hierarchical-support.war`: contiene il supporto per le IMS/MFS transazioni e utilizza `application-jhdb.yaml` come configurazione

Abilita le proprietà per AWS Transform for mainframe Runtime

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. È possibile utilizzare questa pagina per conoscere le proprietà disponibili per AWS Transform for mainframe Runtime e come abilitarle.

Argomenti

- Notazione YML
- Avvio rapido/Casi d'uso
- Proprietà disponibili per l'applicazione principale
- Proprietà disponibili per applicazioni Web opzionali
- Proprietà disponibili per l'applicazione client
- Configura la registrazione degli accessi alle API

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

# Secret manager configuration for global Redis cache
    aws:
      client:
        gapwalk:
          redis:
            secret: arn:aws:secretsmanager:XXXX

########################################
##### 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
    url : jdbc:postgresql://localhost/bluesam
    username : bluesam
    password : bluesam
    type : org.postgresql.ds.PGSimpleDataSource

########################################
##### Embedded Bluesam configuration #####
########################################
bluesam :
  cache : ehcache
  persistence : pgsql
  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
```

```
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

####################################
#####     SSL configuration    #####
####################################
gapwalk.ssl.enabled : true
gapwalk.ssl.trustStore : "./config/clientkey.jks"
gapwalk.ssl.trustStorePassword : mysslcertifpassword

##################################
##### MQ settings #####
##################################
mq.queues: jmsmq
mq.queues.jmsMQQueueManagers[0].jmsMQQueueManager: QM1
mq.queues.jmsMQQueueManagers[0].jmsMQAppName: Gapwalk
mq.queues.jmsMQQueueManagers[0].jmsMQChannel: DEV.APP.SVRCONN
mq.queues.jmsMQQueueManagers[0].jmsMQHost: localhost
mq.queues.jmsMQQueueManagers[0].jmsMQPort: 1415
mq.queues.jmsMQQueueManagers[0].jmsMQUserid: app
mq.queues.jmsMQQueueManagers[0].jmsMQSSLCipher: "*TLS12ORHIGHER"
mq.queues.jmsMQQueueManagers[1].jmsMQQueueManager: QM2
mq.queues.jmsMQQueueManagers[1].jmsMQAppName: Gapwalk
mq.queues.jmsMQQueueManagers[1].jmsMQChannel: DEV.APP.SVRCONN
mq.queues.jmsMQQueueManagers[1].jmsMQHost: localhost
mq.queues.jmsMQQueueManagers[1].jmsMQPort: 1415
mq.queues.jmsMQQueueManagers[1].jmsMQUserid: app

#########################################################
##### Configuration properties for JMS MQ connection ####
#########################################################
mq.queues.jms.connectionfactory.borrow-connection-timeout: 31 # Timeout, in seconds,
 for borrowing connections from the pool.
mq.queues.jms.connectionfactory.ignore-session-transacted-flag: true # Whether or not
 to ignore the transacted flag when creating a session.
mq.queues.jms.connectionfactory.local-transaction-mode: false # Whether or not to
 include local transactions.
```

```
mq.queues.jms.connectionfactory.maintenance-interval: 62 # The time, in seconds,
 between runs of the pool's maintenance thread.
mq.queues.jms.connectionfactory.max-idle-time: 63 # The time, in seconds, after which
 connections are cleaned up from the pool.
mq.queues.jms.connectionfactory.max-lifetime: 0 # The time, in seconds, that a
 connection can be pooled for before being destroyed. 0 denotes no limit.
mq.queues.jms.connectionfactory.max-pool-size: 1 # The maximum size of the pool. This
 property will be overwritten by mq.queues.jmsMQQueueManagers[N].jmsMQMaxPoolSize
 property (if provided).
mq.queues.jms.connectionfactory.min-pool-size: 1 # The minimum size of the pool.
mq.queues.jms.connectionfactory.reap-timeout: 0 # The reap timeout, in seconds, for
 borrowed connections. 0 denotes no limit.


################################
##### 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

#########################################################
#####  BLOCKSIZE DEFAULT VALUE  ####
#########################################################
#blockSizeDefault: 32760

#####################################
#####  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

########################################
#####  JCL Batch Restart Mechanism  ####
########################################
jcl:
checkpoint:
enabled: false
#expireTimeout: -1
#expireTimeoutUnit: SECONDS # Supported values: java.util.concurrent.TimeUnit
#provider: redis

##############################
#####  System parameters  ####
```

```
##############################
system.date.format:MDY
system.date.separator: "/"

#################################################################
#####  Database metadata schema inclusion configuration  ####
#################################################################
gapwalk:
  database:
    metadata:
       schema-inclusion:
          enabled: false
          schemas:
            global:
              - "SCHEMA1"
              - "SCHEMA2"

##############################
#####  URL configuration  ####
##############################
gapwalk.post.script.mediatype.json: false

########################################
#####  CBLQDA parameters  ####
########################################
gapwalk.cobol.cblqda.enabled: false
----
```

- 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 panoramica esaustiva dei key/values parametri.

> **Note**
>
> Se stai migrando alla versione 5.1.0 o successiva e utilizzi bluesam con il comportamento predefinito: openWarmUp
>
> - Il valore predefinito di è cambiato da a `bluesam.openWarmUp` `true` `false`
> - Per mantenere il comportamento esistente, è necessario impostare in modo esplicito `bluesam.openWarmUp=true`

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`system.date.format`	stringa	MDY	Il formato della data di sistema DATFMT.	---
`system.date.separator`	stringa	/	Il separatore di data di sistema DATSEP. I valori possibili sono slash (`/`), dash (`—`), period (`.`), comma (`,`) o blank (``). Il valore deve essere specificato tra virgolette `""`.	5.17.0
`logging.config`	Path	classpath:logback-main.xml	Chiave standard per il riferimento al file di configurazione logback. Sono disponibili anche altre chiavi di registrazione standard.	
`spring.jta.enabled`	booleano	false	Chiave standard. Se la modalità di supporto dell'origine dati non è static-xa	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
			, la configurazione automatica delle transazioni Spring JTA deve essere disabilitata.	
`datasource.jicsDs` + `-driver-class-name` + `-url` + `-username` + `-password` + `-type`	Fonte di dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database Jics. In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato in. the section called "Database JICS"	
`jildb.datasource` + `-driver-class-name` + `-url` + `-username` + `-password`	Fonte di dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database Jics.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jildb.backend`	stringa		Speciifica il tipo di database utilizzato nel backend Jildb, ad esempio Oracle.	
`jildb.record-hold-timeout`	int	10000	Speciifica il periodo di timeout di attesa, in millisecondi, per il meccanismo di conservazione dei record in Jildb.	
`datasource.bluesamDs + -driver-class-name + -url + -username + -password + -type`	Fonte di dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database. Blusam In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato inthe section called "Database Blusam".	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.userspace.pageSize`	int	4096	Speciifica la dimensione della pagina utilizzata per allocare lo spazio di archiviazione per gli spazi utente. La dimensione dello spazio utente in un dato momento è sempre un multiplo di questo numero specificato.	4.10.0
`blu4iv.userspace.store-type`	stringa	memory	Speciifica il tipo di archivio utilizzato per il registro dello spazio utente. I valori supportati sono memory e redis. Se non viene fornito alcun tipo di archivio, per impostazione predefinita viene utilizzata la memoria.	4.10.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.userspace.redis.*`	Proprietà Redis supportate		Specifica le proprietà di configurazione per gli spazi utente redis store, vedi. the section called "Proprietà della cache Redis disponibili"	4.10.0
`blu4iv.userspace.batchThreshold`	int	1	Speciifica il numero di operazioni di aggiornamento o impostazione su uno spazio utente dopo le quali lo spazio utente viene mantenuto in un archivio di dati esterno.	5.75.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.dataaccess.useOldDaoDesign`	booleano	false	Un flag globale che determina se l'applicazione deve utilizzare la vecchia architettura DAO o la nuova. Ciò dipende dal fatto che i DAO siano stati generati utilizzando un design vecchio o nuovo.	
`bluesam.disabled`	booleano	false	Se disabilitarlo Blusam completamente.	
`bluesam.cache`	stringa		Se non è impostata, la Blusam cache non verrà utilizzata. I valori possibili (implementazioni della cache) sono ehcache e redis (). the section called "Proprietà della cache Redis"	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`bluesam.openWarmUp`	booleano	false	Speciifica se memorizzare nella cache eventuali set di dati bluesam durante l'apertura. Avviso: Breaking Change nella versione 5.1.0 - Il valore predefinito è cambiato da **`true`** a. **`false`**	5.1.0
`bluesam.fileList`	stringa []		Speciifica l'elenco dei set di dati bluesam da memorizzare nella cache durante l'apertura. Nota: - Questo parametro è efficace solo quando è impostato su. **`bluesam.openWarmUp false`**	5.1.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`bluesam.maxBluesamDisablingThreadpoolSize`	numero	10	Specifica la dimensione massima del threadpool utilizzata per disabilitare i set di dati bluesam per l'elaborazione in batch.	4.5.0
`bluesam.bluesamStatusPollingInterval`	numero	1000	Speciifica il tempo (in millisecondi) di attesa tra ogni iterazione durante il polling di bluesam status per verificare le attività online.	4.5.0
`bluesam.maxBluesamStatusPollingRetry`	numero	3	Speciifica il numero massimo di tentativi quando il polling di bluesam status ha esito negativo.	4.5.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`bluesam.checkBluesamStatus`	booleano	false	Speciifica se controllare o meno lo stato del set di dati bluesam prima di accedervi.	4.5.0
`bluesam.prefetchWindowSize`	int	1000	Specifica il numero di record da prerecuperare e memorizzare nella cache per l'ottimizzazione della cache Redis. Questa proprietà è efficace solo quando si utilizza la cache Redis.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`bluesam.prefetchWindowMultiples`	int	5	Specificate il moltiplicatore per la dimensione della finestra di prefetch per determinare la capacità massima della finestra di prefetch. La capacità massima viene calcolata come. `prefetchWindowSize × prefetchWindowMultiples` Questa proprietà è efficace solo quando si utilizza la cache Redis.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`spring.aws.client.bluesam.redis.secret`	stringa	null	Specifica l'ARN segreto delle credenziali per la cache Bluesam Redis, vedere. the section called "AWS I segreti di Transform for mainframe Runtime"	
`spring.aws.client.bluesam.locks.redis.secret`	stringa	null	Specifica l'ARN segreto delle credenziali per Bluesam blocca la cache Redis, vedi. the section called "AWS I segreti di Transform for mainframe Runtime"	
`forcedDate`	stringa		Impone la data alla data fornita, se presente.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`clcommand.rtvmbrd.hasHeader`	stringa	true	Specificate se i file contengono intestazioni che non devono essere incluse nel conteggio delle righe.	
`frozenDate`	booleano	true	Specifica se congelare la data. Si applica solo se `forcedDate` è impostata anche.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`externalSort.threshold`	dimensione dei dati (esempio: 12 MB)		La soglia di ordinamento: determina quando il sistema deve passare dall'ordinamento in memoria a quello di unione esterna utilizzando file temporanei sul file system. Se non viene specificata (default=`null`), la soglia viene impostata sulla metà della memoria disponibile per ogni nuova operazione di ordinamento. Per stimare la memoria disponibile, viene eseguita una chiamata a System.gc ().	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blockSizeDefault`	numero	32760	La dimensione del blocco predefinita da utilizzare per i byte BDW.	
`jics.parameters.datform`	stringa	MMDDYY	Il modulo della data.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.initList`	stringa		L'elenco JICS di inizializzazione, separato da virgole. Se presente, definisce i nomi degli elenchi separati da virgole da attivare all'avvio di Apache Tomcat tra gli elenchi CICS. `$UUU,DFH$ IVPL,PEZ1` Valore di esempio:. 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.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.parameters.applid`	stringa	VELOCITÀ	Applicato per identificare l'applicazione in JICS (almeno 4 caratteri, nessuna lunghezza massima).	
`jics.parameters.sysid`	stringa	CICS	L'identificazione del sistema (SYSID).	
`jics.parameters.eibtrmid`	stringa	TERMINE	L'identificatore del terminale (massimo 4 caratteri, minimo 1).	
`jics.parameters.userid`	stringa		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.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.parameters.username`	stringa	IL MIO NOME UTENTE	Il nome utente (massimo 10 caratteri, minimo 1).	
`jics.parameters.netname`	stringa	IL MIO NOME DI RETE	Il nome della rete (massimo 8 caratteri, minimo 1).	
`jics.parameters.opid`	stringa	XXX	L'identificazione dell'operatore a 3 caratteri.	
`jics.parameters.jobname`	stringa	NOME DEL LAVORO	Il nome del lavoro.	
`jics.parameters.sysname`	stringa	SYSNAME	Il nome AS400 del sistema (sysname).	
`jics.parameters.cwa.length`	numero	0	La lunghezza dell'area di lavoro comune (CWA).	
`jics.parameters.charset`	stringa	CP037	Set di caratteri JICS utilizzato a livello globale.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.parameters.tsqimpl`	stringa	bluesam	Implementazione JICS Temporary Storage Queue (TSQ) (i valori consentiti sono//) `bluesam` `memory` `redis`	
`jics.queues.ts.redis.*`	Proprietà Redis supportate		Specifica le proprietà di configurazione per il server Redis JICS TS Queues, vedi. the section called "Proprietà Redis supportate"	
`spring.aws.client.jics.queues.ts.redis.secret`	stringa	null	Specificare l'ARN segreto delle credenziali per il server Redis JICS TS Queues, vedere. the section called "AWS I segreti di Transform for mainframe Runtime"	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`lockTimeout`	numero	500	Il timeout di blocco, in millisecondi.	
`sqlCodePointShift`	numero		Opzionale. Lo spostamento dei punti del codice sql. Sposta il codepoint per i caratteri di controllo che potremmo incontrare durante la migrazione e dei dati RDBMS legacy a un RDBMS moderno. Ad esempio, puoi specificare che corrisponda al carattere Unicode. `384` `\u0180`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`sqlIntegerOverflowAllowed`	booleano	false	Speciifica se consentire l'overflow di numeri interi SQL, ovvero se è consentito inserire valori più grandi nella variabile host.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`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`	stringa	`/reports`	Il percorso di output del report.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`server.servlet.session.tracking-modes`	stringa	biscotto	Specifica le modalità di tracciamento della sessione da utilizzare. I valori possibili sono `urlcookie`,`ssl`. È possibile specificare più valori separati da virgole. Nota: quando si esegue la distribuzione su Tomcat, è necessario impostarlo `cookies=false` in context.xml perché Tomcat ha la precedenza per la configurazione della modalità di tracciamento.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`server.servlet.session.timeout`	stringa	30 m	Speciifica la durata del timeout della sessione. Esempi: `60s` (60 secondi), `4m` (4 minuti), `4h` (4 ore).	
`spring.session.store-type`	stringa	nessuno	La cache della sessione per ambienti ad alta disponibilità. I valori possibili sono `none` o `redis`. L'impostazione predefinita è `none`.	
`stopExecutionWhenProgNotFound`	booleano	true	Speciifica se interrompere l'esecuzione se non viene trovato un programma. Se impostato su `true`, interrompe l'esecuzione se non viene trovato un programma.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`forceHR`	booleano	false	Speciifica se utilizzare Human Readable SYSPRINT, sulla console o sull'output del file.	
`rollbackOnRTE`	booleano	false	Specifica se eseguire il rollback della transazione implicita dell'unità di esecuzione sulle eccezioni di runtime.	
`sctThreadLimit`	Long	5	Il limite dei thread per l'attivazione degli script.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`dataSimplifier.onInvalidNumericData`	stringa	rifiutare	Come reagire quando si decodificano dati numerici non validi. I valori consentiti sono`reject`/`/tolerates paces` . `tolerates paceslowvalues toleratemost` L'impostazione predefinita è`reject`.	
`dataSimplifier.numproc`	stringa	PDF	specificazione del compilatore numproc. I valori consentiti sono//PFD/. `NOPFD MIG` Il valore predefinito è `PFD`.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`dataSimplifier.doubleFloatingType`	stringa	IEEE_754	Il tipo a doppia virgola mobile è il formato usato per codificare e decodificare un numero a virgola mobile. I valori consentiti sono`IEEE_754`/HF il formato esadecimale a virgola mobile). Il valore predefinito è `IEEE_754`.	4.7.0
`filesDirectory`	stringa		La directory per i file batch. input/output	
`ims.messages.extendedSize`	booleano	false	Speciifica se impostare la dimension e estesa dei messaggi IMS.	
`defaultKeepExistingFiles`	booleano	false	Speciifica se impostare il valore precedente predefinito del set di dati.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.db.ddlScriptLocation`	stringa		La posizione dello script DDL di Jics. Consente di avviare lo schema del database Jics utilizzando uno script.sql. Vuoto per impostazione predefinita. Ad esempio, `./jics/sql/jics.sql` .	
`jics.db.schemaTestQueryLocation`	stringa		Posizione del file sql che dovrebbe contenere una query univoca che restituisce il numero di oggetti nello schema jics (se presente).	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.db.dataScriptLocation`	stringa		Definisce il percorso degli script SQL utilizzati per inizializzare il database JICS. Accetta un elenco separato da virgole di file e directory, che consente di specificare più script e cartelle.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.db.dataTestQueryLocation`	stringa		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 `jics.db.dataScriptLocation` lo script, altrimenti il caricamento del database verrà saltato.	
`jics.data.dataJsonInitLocation`	stringa			
`jics.xa.agent.timeout`	numero			

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`query.useConcatCondition`	booleano	false	Speciifica se la condizione della chiave è creata o meno mediante la concatenazione di chiavi.	
`system.qdecfmt`	stringa			
`disposition.checkexistence`	booleano	false	Speciifica se rilasciare un controllo sull'esistenza dei file per Dataset with DISP SHR o OLD.	
`useControlMVariable`	booleano	false	Specifica se utilizzare la specifica Control-M per la sostituzione delle variabili.	
`card.encoding`	stringa	CP1145	Codifica delle carte: da utilizzare con. `useControlMVariable`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mapTransfo.prefixes`	stringa	&,@,%%	Elenco di prefissi da utilizzare per la trasformazione delle variabili ControlM. Ciascuno separato da una virgola.	
`checkinputfilesize`	booleano	false	Specifica se rilasciare un segno di spunta se la dimensione del file è un multiplo della dimensione del record.	
`stepFailWhenAbend`	booleano	true	Specifica se alzare un abend se un passaggio ha esito negativo o completa l'esecuzione.	
`returnHttp5xxResponseForFailedSyncJob`	booleano	false	Speciifica se restituire il codice di risposta HTTP della serie 500 per l'esecuzione non riuscita di un processo sincronizzato.	4.10.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`bluesam.fileLoading.commitInterval`	numero	100000	L'intervallo di commit di bluesam.	
`uppercaseUserInput`	booleano	true	Speciifica se l'input dell'utente deve essere in maiuscolo.	
`jhdb.lterm`	stringa		Consentono di forzare un ID di terminale logico comune nel caso di un'emulazione IMS. Se non è impostato, viene utilizzato SessionID.	
`jhdb.identificationCardData`	stringa		Utilizzato per codificare alcuni «dati della carta di identificazione dell'operatore» nel campo MID indicato dal parametro CARD. Vuoto per impostazione predefinita, nessuna restrizione di input.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`encoding`	stringa	ASCII	La codifica utilizzata nei progetti (non nei file groovy). Prevede una codifica `CP1047` valida,,,... `IBM930` `ASCII` `UTF-8`	
`cl.configuration.context.encoding`	stringa	CP297	La codifica dei file CL. Prevede una codifica validaCP1047,,IBM `ASCII` `UTF-8` Il valore predefinito è `CP297`	
`cl.zonedMode`	stringa	EBCDIC_STRICT	La modalità per la codifica o la decodifica dei comandi del linguaggio di controllo (CL). I valori consentiti sonoEBCDIC_STRICT //EBCDIC_MDIFIED . `AS400`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`ims.programs`	stringa		Elenco dei programmi IMS da utilizzare. Separare ogni parametro con un punto e virgola (`;`) e ogni transazione con una virgola (`,`). Ad esempio:`PCP008,PCT008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;`	
`jhdb.configuration.context.encoding`	stringa	CP297	La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida,,, ... `CP1047` `IBM930` `ASCII` `UTF-8`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jhdb.metadata.extrapath`	stringa	file:. /configurazione/	Un parametro di configurazione che specifica una cartella principale e aggiuntiva, specifica per il runtime, per le cartelle psbs e dbds.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jhdb.checkpointPersistence`	stringa	nessuno	La modalità di persistenza del checkpoint. I valori consentiti sononone//add. end Si usa add per mantenere i checkpoin t quando ne viene creato uno nuovo e aggiunto al registro. Usa per end persister e il checkpoin t alla chiusura del server. Qualsiasi altro valore disabilita la persistenza. Tieni presente che ogni volta che viene aggiunto un nuovo checkpoin t al registro, tutti i checkpoin t esistenti verranno serializzati e il file verrà cancellato. Non è un'aggiunta ai dati esistenti	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
			nel file. Quindi, a seconda del numero di checkpoint, può avere alcuni effetti sulle prestazioni.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jhdb.checkpointPath`	stringa	file:. /configurazione/	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. `none` Nota che solo i dati del checkpoint (scriptID, StepID, posizione del database e area del checkpoint) sono interessa	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
			ti da questo backup.	
`jhdb.navigation.cachenexts`	numero	5000	La durata della cache (in millisecondi) utilizzata nella navigazione gerarchica per un RDBMS.	
`jhdb.use-db-prefix`	booleano	true	Speciifica se abilitare un prefisso di database nella navigazione gerarchica per un RDBMS.	
`jhdb.query.limitJoinUsage`	booleano	true	Speciifica se utilizzare il parametro limit join usage sui grafici RDBMS.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`taskExecutor.corePoolSize`	numero	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.	
`taskExecutor.maxPoolSize`	numero	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	Tipo	Valore predefinito	Description	Versione di rilascio
`taskExecutor.queueCapacity`	numero	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) `taskExecutor.maxPoolSize`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`taskExecutor.allowCoreThreadTimeOut`	booleano	false	Speciifica 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).	
`jics.runUnitLauncherPool.enable`	booleano	false	Specifica se attivare il run unit launcher pool in JICS.	
`jics.runUnitLauncherPool.size`	numero	20	La dimensione del pool di avvio dell'unità di esecuzione in JICS.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.runUnitLauncherPool.validationInterval`	numero	1000	L'intervallo tra ogni esecuzione dell'attività che regola la dimensione del pool.	
`jics.runUnitLauncherPool.parallelism`	numero	2	Il numero di thread utilizzati per produrre le istanze mancanti nella coda durante l'esecuzione dell'attività di regolazione.	
`context.preconstruct.enable`	booleano	false	Speciifica se attivare la precostruzione del contesto del programma.	
`context.preconstruct.frequencyInMillis`	numero	100	L'intervallo tra ogni esecuzione dell'attività che regola le dimensioni del pool.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`context.preconstruct.parallelism`	numero	5	Il numero di thread utilizzati per produrre le istanze mancanti nella coda durante l'esecuzione dell'attività di regolazione.	
`context.preconstruct.minInstances`	numero	2	Il numero di istanze che verranno create la prima volta che è necessario un contesto.	
`spring.aws.application.credentials`	stringa	null	Carica le AWS credenziali dal file dei profili delle credenziali in JICS.	
`jics.queues.sqs.region`	stringa	eu-west-1	La AWS regione per Amazon Simple Queue Service, utilizzata in JICS.	
`jics.jcl.rt.encoding`	stringa	CP037	La codifica degli script JCL scritti nella coda JICS dedicata.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.jcl.rt.queue`	stringa	JICS	Il nome della coda in cui è possibile scrivere gli script JCL riga per riga in fase di esecuzione.	
`mq.queues.sqs.region`	stringa	eu-west-3	La AWS regione per il servizio AWS SQS MQ.	
`quartz.scheduler.stand-by-if-error`	booleano	false	Speciifica 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.	
`databaseStatistics`	booleano	false	Speciifica se consentire ai costruttori di SQL di raccogliere e visualizzare informazioni statistiche.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`dbDateFormat`	stringa	yyyy-MM-dd	Il formato della data di destinazione del db.	
`dbTimeFormat`	stringa	HH: mm: ss	Il formato dell'ora di destinazione del db.	
`dbTimestampFormat`	stringa	yyyy-MM-dd HH: mm: ss.ssssss	Il formato del timestamp di destinazione del db.	
`dateTimeFormat`	stringa	ISO	dateTimeFormat Descrive come trasferire il tipo di data, ora e ora del database in entità semplificatrici di dati. I valori consentiti sono///`ISOEUREUR`	
`localDateFormat`	stringa		Elenco di formati di data locali. Separa ogni formato con. \	
`localTimeFormat`	stringa		Elenco dei formati dell'ora locale. Separa ogni formato con \	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`localTimestampFormat`	stringa		Elenco dei formati di timestamp locali. Separa ogni formato con. \	
`pgmDateFormat`	stringa	yyyy-MM-dd	Il formato della data e dell'ora.	
`pgmTimeFormat`	stringa	HH.mm.ss	Il formato dell'ora usato per l'esecuzione di pgm (programmi).	
`pgmTimestampFormat`	stringa	yyyy-MM-GD-hh.mm.ss.ssssss	Il formato del timestamp.	
`cacheMetadata`	booleano	true	Specifica se memorizzare nella cache i metadati del database.	
`forceDisableSQLTrimStringType`	booleano	false	Speciifica se disabilitare il taglio di tutti i parametri della stringa sql.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`fetchSize`	numero		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.	
`qtemp.uuid.length`	numero	9	La lunghezza dell'ID univoco di QTEMP.	
`qtemp.dblog`	booleano	false	Se abilitare la registrazione del database QTEMP.	
`qtemp.cleanup.threshold.hours`	numero	0	Per specificare quando è abilitato `qtemp.dblog` . La durata della partizione db (in ore).	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`invalidDataTolerence`	booleano	true	Speciifica se i dati non validi sono tollerati per il tipo compresso .	
`program.timeout`	numero	-1	Specifica un timeout per qualsiasi esecuzione in secondi. program/transaction Trascorso questo tempo, il sistema proverà a interrompere il programma.	
`gapwalk.line.separator`	stringa	null	Specifica il tipo di separatore di riga in gapwalk. I valori consentiti sono WIN (CRLF)/UNIX (LF)/LINUX (LF). Gli altri valori vengono ignorati e viene utilizzata la proprietà System line.separator.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`enableActivePgmIdCache`	booleano	false	Speciifica se abilitare la cache locale dell'ID del programma attivo. Utilizzate con attenzione questa funzionalità perché le risorse JICS possono essere condivise tra programmi e utenti. Tali risorse possono essere modificate esternamente da qualsiasi amministratore e la cache locale installata potrebbe essere invalidata.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.queues.default.syncpoint`	booleano	false	Speciifica il comportamento predefinito per i comandi MQ PUT quando non sono impostati né MQPMO_SYNCPOINT né MQPMO_NO_SYNCPOINT. Quando è impostato su true, agisce come se i messaggi NON venissero salvati direttamente durante il comando PUT. `MQPMO_SYNCPOINT` Quando è impostato su false, agisce come se `MQPMO_NO_SYNCPOINT` i messaggi venissero salvati direttamente durante il comando PUT.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`dataSimplifier.byteRangeBoundsCheck`	booleano	false	Se impostato su true, assicura che non ByteRange venga creato nessun valore con valori errati. Il valore predefinito è false.	
`file.stdoutIntoLogger`	booleano	false	Speciifica se abilitare la scrittura su logger anziché sul flusso di output di sistema predefinito in default `SYSPRINT` and files. `SYSPUNCH`	
`tempFilesDirectory`	stringa	null	Speciifica il nome della posizione della cartella dei file temporanei generati.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`cleanTempFilesDirectoryAtStartup`	booleano	true	Speciifica se eliminare il contenuto della cartella dei file temporanei all'avvio dell'applicazione.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`tempFolderPattern`	stringa	null	Speciifica uno schema che verrà utilizzato per creare dinamicamente il nome della cartella temporanea in base alle seguenti informazioni predefinite e personalizzabili. HOST: il nome dell'host. JOBID: l'ID del lavoro. HASHCODE: il codice hash del contesto lavorativo. TIMESTAMP: lo schema da utilizzare per ottenere il timestamp. Il nome di destinazione della cartella temporanea è TMP_DIR_	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
			{}. tempFolderPattern Ad esempio, nel caso dello schema seguente, il nome inizierà con l'ID del lavoro e terminerà con il «timestamp» tempFolderPattern: JOBID, host=XXXXX, HASHCODE, timestamp =YYYYMMDDHHMMSS. Se la proprietà non `tempFolderPattern` viene aggiunta al file YAML o è vuota, il nome della cartella temporanea sarà «TMP_DIR_» + this.hashCode () (DefaultJobContext).	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`database.cursor.raise.already.opened.error`	booleano	false	Speciifica se abilitare la generazione dell'errore SQLCODE 502 quando si apre un cursore già aperto.	
`jics.spool.smtp.hostname`	stringa	null	Speciifica l'host del server SMTP. Ad esempio: `smtp.xxx.com`	
`jics.spool.smtp.port`	stringa	null	Speciifica la porta del server SMTP. Esempio: 25	
`jics.spool.smtp.password`	stringa	null	Speciifica la password di accesso del server SMTP.	
`jics.spool.smtp.username`	stringa	null	Speciifica il nome utente del server SMTP.	
`jics.spool.smtp.debug`	booleano	false	Speciifica la modalità di debug per il server SMTP.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk-application.security`	stringa	`disabled`	Attiva la configurazione di sicurezza globale (autenticazione XSS, CORS, CSRF, OAUTH...). I valori consentiti sono `disabled` e `enabled`.	
`gapwalk-application.security.identity`	stringa	null	Metodo di autenticazione globale. Il valore consigliato è`oauth`. I valori consentiti sono `json` e `oauth`. Questa opzione è obbligatoria quando `gapwalk-application.security` è`enabled`.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk-application.security.issuerUri`	stringa	null	L'URI dell'emittente del provider di identità (IdP). Questa opzione è obbligatoria quando `gapwalk-application.identity` è. `oauth`	
`gapwalk-application.security.allowedOrigins`	stringa []	`null`	L'elenco delle origini da consentire. Questa opzione deve `gapwalk-application.identity` essere impostata su`oauth`.	
`gdgDirectoryPath`	stringa	`output/gdg`	Il percorso della directory GDG è la directory in cui sono archiviati i file gdg.	4.6.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk-application.security.customAllowedHeaders`	stringa []	null	L'elenco delle intestazioni personalizzate da consentire. Questa opzione deve `gapwalk-application.identity` essere impostata su. `oauth`	4.8.0
`listOfMDCUrlParams`	stringa	`null`	L'elenco dei parametri URL da aggiungere all'MDC durante l'esecuzione dello script batch Groovy.	
`gapwalk-application.security.claimGroupName`	stringa	`cognito:groups`	L'attributo claim che contiene l'elenco di tutti i gruppi a cui appartiene un utente. `cognito:groups` Utilizzalo per Amazon Cognito o qualsiasi altra stringa per un IdP esterno.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk-application.security.userAttributeName`	stringa	`username`	Il nome dell'attributo claim utilizzato per identificare una richiesta dell'utente. Utilizzalo `username` per Amazon Cognito, `preferred_username` per Keycloak o qualsiasi altra stringa per un IdP esterno.	
`gapwalk-application.defaultSuperAdminUserName`	stringa	`sadmin`	Quando `gapwalk-application.security` è disabilitato, specifica il nome utente privilegiato locale predefinito.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk-application.defaultSuperAdminUserPwd`	stringa	`sadmin`	Quando `gapwalk-application.security` è disabilitato, specifica la password di super utente locale predefinita.	
`gapwalk-application.security.filterURIs`	stringa	`disabled`	Attiva la configurazione del filtro. URIs I valori consentiti sono `disabled` e `enabled`.	
`gapwalk-application.security.blockedURIs`	stringa []	null	L'elenco di URIs da bloccare. Questa opzione è obbligatoria quando `gapwalk-application.security.filterURIs` è`enabled`.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jics.redis.*`	Proprietà Redis supportate		Specifica le proprietà di configurazione per il Connection Factory del server JICS Redis, vedere. the section called "Proprietà Redis supportate"	
`spring.aws.client.jics.redis.secret`	stringa	null	Specifica l'ARN segreto delle credenziali per la fabbrica di connessioni al server JICS Redis, vedere. the section called "AWS I segreti di Transform for mainframe Runtime"	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jcl.checkpoint.enabled`	booleano	false	Speciifica se il meccanismo di checkpoint JCL è abilitato per consentire il riavvio del lavoro. I checkpoint JCL vengono creati e salvati nel registro in memoria all'inizio di ogni passaggio o chiamata principale del programma. Tutti i checkpoint a livello di fase vengono mantenuti alla fine del processo, se è definito il provider di persistenza.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jcl.checkpoint.expireTimeout`	numero	-1	Speciifica la durata del periodo di conservazione dei checkpoint JCL nel provider di persistenza o nel registro in memoria.	
`jcl.checkpoint.expireTimeoutUnit`	stringa	SECONDI	Speciifica l'unità di durata temporale per la `jcl.checkpoint.expireTimeout` proprietà. Valori costanti enum supportati: java.util.concurrent. TimeUnit.	
`jcl.checkpoint.provider`	stringa	null	Specifica il provider di persistenza del meccanismo di checkpoint JCL. I valori consentiti sono. `redis`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`jcl.checkpoint.redis.*`	Proprietà Redis supportate		Specifica le proprietà di configurazione per il provider di persistenza REDIS del meccanismo di checkpoint JCL, vedi. the section called "Proprietà Redis supportate"	
`jcl.nullJclEof`	booleano	true	Speciifica come vengono gestite le istruzioni JCL nulle (//) durante l'esecuzione del lavoro. Se impostato su true,//termina immediatamente il processo (modalità EOF). Se false,//viene considerata un'istruzione nulla e l'esecuzione continua (modalità IGNORE).	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`job.default.encoding`	stringa	CP1047	Specifica la codifica predefinita utilizzata per inizializzare la memorizzazione a livello di processo. Ad esempio, gli switch nell'applicazione BLU4 IV utilizzano questa codifica per l'inizializzazione a zeri.	4.9.0
`spring.aws.client.jcl.checkpoint.redis.secret`	stringa	null	Speciifica l'ARN segreto delle credenziali per il provider di persistenza Redis del meccanismo di checkpoint JCL, vedi. the section called "AWS I segreti di Transform for mainframe Runtime"	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.ssl.enabled`	booleano	false	Indicato per impostare le seguenti `gapwalk.ssl.*` proprietà sulle proprietà correnti del sistema JVM se non sono già impostate all'avvio dell'applicazione.	
`gapwalk.ssl.trustStore`	stringa	null	Imposta il valore sulla proprietà di sistema `javax.net.ssl.trustStore` se non è già impostato all'avvio dell'applicazione.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.ssl.trustStorePassword`	stringa	null	Impostate il valore sulla proprietà di sistema `javax.net.ssl.trustStorePassword` se non è già configurata all'avvio dell'applicazione. In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato in[the section called "Gestore segreto per le impostazioni delle password SSL"].	
`gapwalk.ssl.trustStoreType`	stringa	null	Imposta il valore sulla proprietà di sistema `javax.net.ssl.trustStoreType` se non è già configurata all'avvio dell'applicazione.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.ssl.keyStore`	stringa	null	Imposta il valore sulla proprietà di sistema `javax.net.ssl.keyStore` se non è già configurata all'avvio dell'applicazione.	
`gapwalk.ssl.keyStorePassword`	stringa	null	Imposta il valore sulla proprietà di sistema `javax.net.ssl.keyStorePassword` se non è già configurata all'avvio dell'applicazione. In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato in[the section called "Gestore segreto per le impostazioni delle password SSL"]().	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.queues`	stringa	`sqs`	Speciifica quale queue brocker supportato da utilizzare tra Amazon `sqs` SQS, Rabbit MQ locale o `rabbitmq` IBMMQ locale. `jms`	
`mq.queues.jmsMQQueueManagers[N]`			In caso `mq.queues` `jms` affermativo, consente di specificare un elenco di connessioni IBM MQ. `mq.queues.jmsMQQueueManagers[0]` per la prima connessione, `mq.queues.jmsMQQueueManagers[1]` per la seconda e così via.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.queues.jmsMQQueueManagers[N].jmsMQQueueManager`	stringa	null	Il nome del gestore di code IBMMQ.	
`mq.queues.jmsMQQueueManagers[N].jmsMQAppName`	stringa	null	Il nome dell'applicazione IBMMQ.	
`mq.queues.jmsMQQueueManagers[N].jmsMQChannel`	stringa	null	Il nome del canale IBMMQ.	
`mq.queues.jmsMQQueueManagers[N].jmsMQHost`	stringa	null	Il nome host IBMMQ.	
`mq.queues.jmsMQQueueManagers[N].jmsMQPort`	numero	null	La porta IBMMQ.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.queues.jmsMQQueueManagers[N].jmsMQUserid`	stringa	null	Il nome utente IBMMQ.	
`mq.queues.jmsMQQueueManagers[N].jmsMQPassword`	stringa	null	La password utente IBMMQ. In alternativa, l'uso dei AWS segreti è fortemente e consigliato, come spiegato in. the section called "Gestore segreto per le impostazioni delle password di IBM MQ"	
`mq.queues.jmsMQQueueManagers[N].jmsMQMaxPoolSize`	numero	0	La dimensione massima del pool IBMMQ. Con 0, è abilitato un numero infinito di connessioni fisiche.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.queues.jmsMQQueueManagers[N].jmsMQSSLCipher`	stringa	null	La suite di crittografia SSL IBMMQ. Un esempio potrebbe essere. `"*TLS12OR HIGHER"` Per maggiori dettagli, consulta la documentazione ufficiale TLS CipherSpecs e CipherSuites nelle classi IBM MQ per JMS.	
`mq.queues.non.jms.client`	booleano	false	Indica se il client di destinazione a cui inviare messaggi non è JMS. Il formato MQ nativo verrà utilizzato per client non JMS mentre il RFH2 formato verrà utilizzato per JMS.	4.5.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mq.connection.pool.share`	booleano	false	Indica se condividere o ricreare il pool di connessioni JMS con lo stesso Queue Manager.	4.9.0
`mq.queues.rabbitMQHost`	stringa	null	Il nome host di Rabbit MQ.	
`mq.queues.rabbitMQVirtualHost`	stringa	null	L'hostname virtuale Rabbit MQ.	
`mq.queues.rabbitMQPort`	numero	null	La porta Rabbit MQ.	
`mq.queues.rabbitMQUsername`	stringa	null	L'utente Rabbit MQ.	
`mq.queues.rabbitMQPassword`	stringa	null	La password di Rabbit MQ.	
`mf.runtime.switch.N`	booleano	true	Abilita l'inserimento nullo per i file sequenziali di linea MF Nature.	4.4.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`mf.runtime.switch.T`	booleano	false	Consente l'inserimento di caratteri di tabulazione nei file sequenziali di linee MF Nature.	4.4.0
`gapwalk.database.support.useSavePointToRestoreFail`	booleano	false	Consente il ripristino delle transazioni in caso di errore utilizzando i savepoint nelle query di inserimento. L'attivazione di questa proprietà può influire sulle prestazioni del database. È possibile sovrascrivere questa impostazione per query specifiche utilizzando la configurazione di query-to-database mappatura.	4.6.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`xmlGeneratorPrettyFormat`	booleano	true	Determina lo stile di formattazione dell'output XML generato all'interno della classe. XmlGenerator Se impostato su`true`, l'output XML viene formattato in una struttura rientrata leggibile dall'uomo (stampata in modo carino). Se impostato su`false`, l'output è in un formato compatto a riga singola senza spazi bianchi aggiuntivi.	4.8.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.database.metadata.schema-inclusion.enabled`	booleano	false	Abilita una funzionalità di ottimizzazione della memoria che controlla quali schemi di database vengono memorizzati nella cache da un'applicazione. Questa configurazione aiuta a ridurre l'ingombro della memoria mantenendo al contempo la stabilità del sistema gestendo in modo intelligente i metadati dello schema. Include automaticamente alcuni schemi di sistema di database comuni che potrebbero essere necessari per il corretto funzionamento.	4.9.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.database.metadata.schema-inclusion.schemas.global`	elenco <String>	null	Definisce un elenco di schemi che devono essere memorizzati nella cache.	4.9.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`sqlConnectionAutoCommitEnabled`	booleano	true	Abilita il commit automatico della connessione corrente durante le operazioni di ripristino della connessione. Se abilitato, esegue automaticamente il commit della connessione corrente prima di passare a una nuova connessione all'origine dati in ambienti non XA per prevenire l'instabilità della connessione. Se disattivata, registra un messaggio di avviso sui potenziali problemi di connessione durante le operazioni di ripristino.	4.10.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.post.script.mediatype.json`	booleano	false	Speciifica se l'`Content-Type` intestazione della risposta da endpoint `/post/script/{scriptId:.+}` è impostata su `application/json`, a seconda del valore di una variabile booleana. Se impostato su false, l'impostazione predefinita è. `text/plain; charset=utf-8`	
`startDefaultJob`	stringa	`false`	Specificare se inizializzare il processo predefinito alla prima transazione.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`gapwalk.cobol.cblqda.enabled`	booleano	false	Abilita una funzionalità CBLQDA che controlla l'allocazione dinamica dei file QSAM durante un'istruzione OPEN. Questa configurazione aiuta ad allocare dinamicamente i file temporanei dichiarati nel programma, anche se i file non sono definiti nello script JCL.	
`mq.queues.jms.connectionfactory.*`	Proprietà di fabbrica di connessione Atomikos supportate		Specifica le proprietà di configurazione per il pool di connessioni JMS MQ.	

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, ad esempio z/OS, AS/400, or IMS/MFS. The following tables contain lists of the available key/value i parametri per la configurazione di ciascuna applicazione Web opzionale.

gapwalk-utility-pgm.guerra

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

Questa tabella fornisce una panoramica esaustiva dei key/values parametri di questa applicazione.

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`logging.config`	Path	classpath: logback-utility.xml	Chiave standard per il riferimento al file di configurazione logback. Sono disponibili anche altre chiavi di registrazione standard.	
`spring.jta.enabled`	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.	
`spring.datasource.primary.jndi-name`	stringa	jdbc/primario	Il nome JNDI (Java Naming And Directory Interface) per l'origine dati principale, se si utilizza JNDI.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`primary.datasource -driver-class-name -url -username -password`	Fonte dati Spring standard con sottochiavi		Contiene le informazioni di connessione per il database dell'applicazione, se non si utilizza JNDI. Deve avere la stessa configurazione del file YAML dell'applicazione modernizzata. In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato in. the section called "Database client"	
`encoding`	stringa	ASCII	La codifica utilizzata nei programmi di utilità. Prevede una codifica validaCP1047,,IBM `ASCII` `UTF-8`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`sysPunchEncoding`	stringa	ASCII	Il set di caratteri di codifica syspunch. Prevede una codifica valida,,. `CP1047` `IBM930` `ASCII` `UTF-8`	
`systin.encoding`	stringa	ASCII	Il set di caratteri di codifica del set di dati del file SYSTIN. Prevede una codifica `CP1047` valida,,. `IBM930` `ASCII` `UTF-8`	4.5.0
`zonedMode`	stringa	EBCDIC_STRICT	La modalità per codificare e o decodificare i tipi di dati suddivisi in zone. I valori consentiti sono//`EBCDIC_STRICT` . `EBCDIC_MODIFIED_AS400`	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`idcams.encoding.forced`	stringa		La codifica utilizzata nel programma di utilità IDCAMS. Prevede una codifica `CP1047` valida,,. `IBM930` `ASCII` `UTF-8`	4.4.0
`unload.chunkSize`	numero	0	Dimensione del blocco utilizzata per l'utilità di scarico.	
`unload.computeRecordSizeIfNull`	booleano	false	Determina se calcolare la dimensione del record se non specificata. Se specificato, il valore rimane invariato.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`unload.sqlCodePointShift`	numero	0	Il codice SQL pointshift per l'utilità unload. Esegue il processo di cambio dei caratteri . Richiesto quando il database di destinazione DB2 è Postgresql.	
`unload.columnFiller`	stringa	spazio	Il riempitivo di colonne dell'utilità unload.	
`unload.varCharIsNull`	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	Tipo	Valore predefinito	Description	Versione di rilascio
`unload.useDatabaseConfiguration`	booleano	false	Speciifica se utilizzare la configurazione di data o ora da application-main.yml nell'utilità unload.	
`unload.format.date`	stringa	MM/dd/yyyy	Se abilitato, `unload.useDatabaseConfiguration` il formato della data da utilizzare nell'utilità di scarico.	
`unload.format.time`	stringa	H.mm.ss	Se `unload.useDatabaseConfiguration` abilitato, il formato dell'ora da utilizzare nell'utilità di scaricamento.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`unload.format.timestamp`	stringa	yyyy-MM-dd-HH.mm.ss.ssssss	Se abilitato, `unload.useDatabaseConfiguration` il formato del timestamp da utilizzare nell'utilità di scaricamento.	
`unload.nbi.whenNull`	esadecimale	6F	Il valore Null Byte Indicator (NBI) da aggiungere quando il valore del database è nullo.	
`unload.nbi.whenNotNull`	esadecimale	00	Il valore Null Byte Indicator (NBI) da aggiungere quando il valore del database non è nullo.	
`unload.nbi.writeNullIndicator`	booleano	false	Speciifica se scrivere l'indicatore null nel file di output di scaricamento.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`unload.bmc.useInto`	booleano	false	Speciifica se gestire la parola chiave di controllo INTO bmc per l'utilità unload.	
`unload.fetchSize`	numero	0	Consente di regolare la dimensione del recupero quando si gestiscono i cursori nell'utilità unload.	
`unload.noPad`	booleano	true	Indica che i campi con caratteri a lunghezza variabile (VARCHAR) devono essere scaricati senza alcuna imbottitura fino alla lunghezza massima.	4.5.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`treatLargeNumberAsInteger`	booleano	false	Speciifica se trattare i numeri grandi come. `Integer` Per impostazione predefinita, vengono `BigDecimal` trattati come tali.	
`load.batchSize`	numero	0	La dimensione del batch dell'utilità di caricamento.	
`load.format.localDate`	stringa	gg.mm.aaaa\\ aaaa-MM-gg dd/MM/yyyy	Il formato di data locale dell'utilità di caricamento da utilizzare.	
`load.format.localTime`	stringa	HH:mm:ss\ hh.mm.ss	Il formato dell'ora locale dell'utilità di caricamento da utilizzare.	
`load.format.dbDate`	stringa	yyyy-MM-dd	Il formato del database dell'utilità di caricamento da utilizzare.	
`load.format.dbTime`	stringa	HH:mm:ss	Il tempo di utilizzo del database dell'utilità di caricamento.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`load.sqlCodePointShift`	numero	0 s	Il codice SQL pointshift per l'utilità di caricamento. Esegue il processo di cambio dei caratteri. Richiesto quando il database di destinazione DB2 è Postgresql.	
`load.applyRollback`	booleano	false	Imposta questo parametro `true` per indicare che desideri che il servizio ripristini le modifiche alla tabella se riscontra un errore durante il caricamento dei dati nel database.	
`forcedDate`	stringa		Impone la data alla data fornita, se presente.	

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`frozenDate`	booleano	true	Specifica se congelare la data. Si applica solo se `forcedDate` è impostata anche.	
`jcl.type`	stringa	mvs	tipo di file.jcl. I valori consentiti sono/`jcl`. `vse` I PRINT/REPRO comandi dell'utilità IDCAMS restituiscono 4 se il file è vuoto per jcl non vse.	
`hasGraphic`	booleano	false	Se l'utilità INFUTILB deve gestire le colonne GRAPHIC. DB2	
`convertGraphicDataToFullWidth`	booleano	true	Speciifica se convertire i dati grafici in formato a larghezza intera.	

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 key/values parametri di questa applicazione.

Chiave	Tipo	Valore predefinito	Description
`logging.config`	Path	classpath:logback-utility.xml	Chiave standard per il riferimento al file di configura zione logback. Sono disponibili anche altre chiavi di registrazione standard.
`spring.jt a.enabled`	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.
`spring.da tasource. primary.jndi- name`	stringa	jdbc/primario	Il nome JNDI (Java Naming And Directory Interface) per l'origine dati principale, se si utilizza JNDI.
`primary.d atasource + -driver-cl ass-name + -url + -username + -password`	Fonte dati Spring standard con sottochiavi		Contiene le informazi oni di connessio ne per il database dell'applicazione, se non si utilizza JNDI. Deve avere la stessa configurazione del file YAML dell'appl icazione modernizz ata.

Chiave	Tipo	Valore predefinito	Description
			In alternativa, l'uso dei AWS segreti è fortemente incoraggiato, come spiegato in. the section called "Database client"
`encoding`	stringa	ASCII	La codifica utilizzata nei programmi di utilità. Prevede una codifica validaCP1047,,IBM930... `ASCII` `UTF-8`
`zonedMode`	stringa	EBCDIC_STRICT	La modalità per codificare o decodificare i tipi di dati suddivisi in zone. I valori consentiti sono//`EBCDIC_STRICT` . `EBCDIC_MODIFIED` `AS400`

Chiave	Tipo	Valore predefinito	Description
`commands-off`	stringa		Elenco di comandi da disattivare, separati da virgole. I valori consentiti sono `PGM_BASIC` `RCVMSGSNDRCVF,CHGVAR,QCL` Utile quando si desidera disabilit are o sovrascri vere un programma esistente. `PGM_BASIC` è un programma AWS Transform for mainframe Runtime specifico progettato per scopi di debug.
`forcedDate`	stringa		Forza l'inserimento della data alla data fornita, se presente.

gapwalk-hierarchical-support.guerra

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

Questa tabella fornisce una visualizzazione esaustiva dei key/values parametri di questa applicazione.

Chiave	Tipo	Valore predefinito	Description
`logging.config`	Path	classpath:logback-utility.xml	Chiave standard per il riferimento al file di configura zione logback. Sono disponibili anche altre

Chiave	Tipo	Valore predefinito	Description
			chiavi di registrazione standard.
`spring.jta.enabled`	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.
`jhdb.configuration.context.encoding`	stringa		La codifica JHDB (Java Hierarchical Database). Prevede una stringa di codifica valida,,,... `CP1047` `IBM930` `ASCII` `UTF-8`

Chiave	Tipo	Valore predefinito	Description
`jhdb.checkpointPersistence`	stringa	nessuno	La modalità di persistenza del checkpoint. I valori consentiti sononone//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.

Chiave	Tipo	Valore predefinito	Description
`jhdb.keepParent`	booleano	false	Se mantenere _parent e _logicalparent durante l'inserimento di IMS. Se impostata su true, la chiamata ISRT IMS inserirà _parent e _logicalp arent nelle colonne del database.
`jhdb.quer y.timeout`	int	-1	Specifica un timeout per l'esecuzione di qualsiasi transazio ne jhdb in secondi. Trascorso questo tempo, il sistema proverà a interromp ere il programma. Non viene impostato alcun timeout se il valore è -1 (impostazione predefinita).
`jhdb.backend`	stringa		Tipo di backend Jhdb. Il valore previsto è `rdbms`
`jhdb.tran saction.s cope.programs`	stringa		elenco di programmi da dichiarare come transazione completa. Separare ogni programma con una virgola (`,`). Ad esempio:`PCP008,PC T008`

Chiave	Tipo	Valore predefinito	Description
`metadata.datasource + -driver-class-name + -url + -username + -password + -type`	stringa	Fonte di dati Spring standard con sottochiavi	Contiene le informazioni sui metadati per il database jhdb, inclusi psbs, dbds, connections. In alternativa, l'uso dei segreti è fortemente incoraggiato. AWS

Proprietà disponibili per l'applicazione client

L'applicazione modernizzata potrebbe richiedere configurazioni di proprietà specifiche per l'applicazione client Spring. Queste proprietà inizializzano i bean dalle classi incluse nei file JAR di runtime. Il `application-profile.yaml` file, in cui viene impostato il valore del profilo durante la generazione dell'applicazione, consente di configurare queste proprietà. La tabella seguente elenca i key/value parametri disponibili per la configurazione dell'applicazione web client che utilizza i bean delle classi incluse nel runtime Gapwalk

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.dtaara.library.disable`	booleano	false	Controlla l'utilizzo della libreria nel contesto delle operazioni dell'area dati. Se impostato su true, l'utilizzo della libreria è disabilitato per le operazioni sull'area dati, ma ciò non influisce sull'utilizzo di QTemp. Se	4.5.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
			impostato su false, la libreria viene considerata quando si eseguono operazioni CRUD per l'area dati.	
`blu4iv.dao.data.max`	numero	10	La dimensione della cache interna per le operazioni DAO di input. Questa cache esiste a livello di programma (istanza singola).	4.9.0
`blu4iv.dao.sort.function`	stringa		Il nome della funzione di ordinamento per il database blu4iv.	4.9.0
`blu4iv.dao.support`	stringa	JPA	Speciifica l'implementazione di accesso ai dati da utilizzare (JPA o JDBC).	4.9.0
`blu4iv.dao.cache.enabled`	booleano	false	Abilita la cache in memoria.	4.8.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.dao.cache.enabledHits`	booleano	false	Aggiungi il tracciamento del numero di richieste alla cache.	4.8.0
`blu4iv.dao.cache.initMaxResults`	numero	10000	La dimensione della cache.	4.8.0
`blu4iv.dao.cache.entries.fileIds`	stringa		L'elenco delle tabelle del database da inserire nella cache.	4.8.0
`blu4iv.dao.cache.entries.programIds`	stringa		Gli identificatori del programma in cui deve essere attivata la cache per le tabelle specifiche. La cache è disponibile per tutti i sottoprogrammi negli stack di esecuzione.	4.8.0

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`blu4iv.librarylist.enabled`	booleano	false	Abilita l'uso dell'elenco delle librerie per risolvere una libreria di file.	4.10.0
`blu4iv.librarylist.libraries`	stringa		L'elenco iniziale delle librerie ordinate da sinistra a destra.	4.10.0

Configura la cache in memoria per le applicazioni AS400

La funzionalità di cache in memoria consente agli utenti di memorizzare nella cache i dati di sola lettura all'interno della memoria, migliorando significativamente le prestazioni per i programmi a uso intensivo di dati. Il sistema di cache ottimizza le prestazioni in tre modi: caricamento dei dati solo al primo accesso, creazione di mappe di accesso ai dati su richiesta e archiviazione efficiente dei dati memorizzati nella cache vicino alla memoria del programma. Questo design riduce al minimo il sovraccarico di memoria massimizzando al contempo i miglioramenti della velocità.

```
#####################################################################
# Cache configuration for read operations called from DAO classes #
#####################################################################
blu4iv:
 dao:
   cache:
     enabled: true
     enabledHits: true
     initMaxResults: 100000
     entries:
     -
       fileIds: [${TABLEA}]
       programIds: [${PROGRAM1}]
     -
       fileIds: [${TABLEB},${TABLEC}]
       programIds: [${PROGRAM2}]
```

```
  librarylist:
         enabled: true
         libraries: ["LIBA","LIBC","LIBE"]
```

Configura la registrazione degli accessi alle API

La funzione di registrazione degli accessi alle API registra informazioni dettagliate sulle richieste API alle applicazioni BAC, JAC e Gapwalk, contribuendo a soddisfare i requisiti di sicurezza tracciando chi ha effettuato l'accesso a cosa e quando.

Chiave	Tipo	Valore predefinito	Description	Versione di rilascio
`api.logging.enabled`	booleano	false	Abilita la registrazione degli accessi alle API per le applicazioni BAC, JAC e Gapwalk. Se abilitato, registra tutte le richieste API con dettagli.	5.0.0

Se abilitato, l'utente può fornire la configurazione del file di logback per registrare i log di accesso all'API nella posizione prescelta utilizzando il nome del logger. `api.access` I log possono includere i seguenti campi MDC:

Campo MDC	Description
`api-date`	La data e l'ora in cui è stata effettuata la richiesta API
`api-duration`	Quanto tempo è stato impiegato per elaborare la richiesta API (millisecondi)
`api-host`	L'indirizzo di origine da cui proviene la richiesta API

Campo MDC	Description
`api-context`	Dove è ospitata l'API
`api-user`	Il nome utente o l'identificatore della persona che effettua la richiesta API
`api-sessionid`	Un identificatore univoco per la sessione di accesso corrente dell'utente
`api-scheme`	Il protocollo di trasferimento utilizzato, ad esempio, HTTP o HTTPS.
`api-name`	A endpoint/URI cui è stato effettuato l'accesso
`api-retcode`	Il codice di stato restituito dopo il completamento della richiesta API
`api-secure`	Il metodo di sicurezza utilizzato per proteggere la chiamata API (ad esempio, Token OAuth2, Anonymous)
`api-method`	Il tipo di richiesta HTTP effettuata (ad esempio, GET, POST, DELETE,...)

Per configurare un file di registro dedicato per i log di accesso alle API, aggiungi quanto segue alla tua configurazione di logback:

```
<!-- File appender -->
<appender name="ApiAccessFile" class="ch.qos.logback.core.FileAppender">
    <file>logs/main/main-api-access.log</file>
    <immediateFlush>true</immediateFlush>
    <encoder>
        <pattern>%date %X{api-scheme} %X{api-host} %X{api-method} %X{api-name} - Return
 code: %X{api-retcode} - Start date: %X{api-date} - Duration: %X{api-duration} ms -
 User: %X{api-user} - SessionId: %X{api-sessionid} - SecureMethod: %X{api-secure} %n</
pattern>
    </encoder>
</appender>

<logger name="api.access" level="info" additivity="false">
    <appender-ref ref="ApiAccessFile" />
    <appender-ref ref="CONSOLE" />
</logger>
```

Esempio di output del registro:

```
2025-06-13 22:24:36,174 http 127.0.0.1 GET /bac/api/services/rest/bluesamservice/
listDataSet - Return code: 200 - Start date: 06-13-2025 22:24:35.863+0200 - Duration:
 308 ms - User: velocity_admin - SessionId:
1X1X1X1X1X1X1X1X1X1X1X1X1X1X1X- SecureMethod: Oauth2
```

Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Puoi usare questo documento per conoscere le cache Redis in AWS Transform for mainframe Runtime, insieme alla configurazione di Gapwalk, alle proprietà Redis supportate e come il file `application-main.yml` può fare riferimento all'ARN segreto per le cache Redis.

Cache Redis in Transform for mainframe Runtime AWS

I server Redis possono essere utilizzati come cache per varie funzionalità dell'applicazione Gapwalk AWS Transform for mainframe, come:

AWS Funzionalità di Transform for mainframe Runtime che utilizzano la memorizzazione nella cache Redis	Description
Blusamcache	Una Blusam cache Redis per leggere i record in modo efficiente, utilizzando una strategia write-behind, per ottimizzare i carichi di lavoro ad alta intensità di scrittura riscontrati sui payload in batch.
Blusamserrature	Una cache per i blocchi distribuiti per set di dati e record.
Catalogo dei set di dati	La cache del set di dati del catalogo.
Cache della sessione	Una cache Redis per HttpSession. La cache memorizza il nome utente, lo stato del dialogo con il frontend Angular e informazioni specifich e sul «dialetto» (BMS, MFS,). AS400

AWS Funzionalità di Transform for mainframe Runtime che utilizzano la memorizzazione nella cache Redis	Description
Tracciatore della sessione	Una cache di sessioni attive con nome utente e session-creation-time informazioni associati.
Cache ICS	Una cache per le definizioni delle risorse JICS.
Code TS	Archiviazione per code TS.
Punto di controllo JCL	Cache del checkpoint JCL.
Blocchi di file Gapwalk	Una cache per i blocchi di file distribuiti per processo.
Blocchi Blu4iv	Archiviazione per i blocchi di registrazione Blu4iv.

Configurazione Redis Gapwalk

La configurazione Redis globale viene utilizzata se `redis` è specificata come meccanismo di memorizzazione nella cache e non viene fornita alcuna configurazione Redis per la funzionalità specifica. Questa configurazione consente di utilizzare la stessa configurazione per più cache Redis contemporaneamente.

Nell'esempio seguente, la cache Blusam dei set di dati e la cache JICS utilizzano la configurazione `gapwalk.redis` (`redis.server1`) perché il loro tipo di cache è impostato su e non sono specificate proprietà Redis implicite in e. `redis` the section called "Definizioni delle risorse JICS" the section called "Definizioni delle risorse JICS" Tuttavia, la cache dei Blusam blocchi utilizzerà una configurazione Redis diversa (`redis.server2`) perché le sue proprietà Redis sono definite in modo esplicito.

```
...

 gapwalk:
   redis:
     hostName: redis.server1
  port: 6379
```

```
...

bluesam:
  # Redis bluesam cache
  cache: redis
  # Redis locks cache
  locks:
    cache: redis
 hostName: redis.server2
 port: 6379
...
# Redis jics cache
jics:
  resource-definitions:
    store-type: redis
...
```

Per abilitare la configurazione globale di Redis, aggiungi la seguente configurazione in. `main-application.yml`

```
gapwalk:
  redis:
    hostName: localhost
    port: 6379
    mode: standalone                        # Optional
    username:                               # Optional
    password: ""                            # Optional
    useSsl: false                           # Optional
    database: 0                             # Optional
    maxTotal: 128                           # Optional
    maxIdle: 128                            # Optional
    minIdle: 16                             # Optional
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
```

```
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

Proprietà Redis supportate

La tabella seguente mostra le proprietà Redis supportate per le cache Redis globali e specifiche su AWS Transform for mainframe Runtime.

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`mode`	No	La modalità di esecuzione Redis.	`standalone | cluster`	`standalone`
`hostname`	Sì	Il nome host o l'indirizzo IP del server Redis.	stringa	null
`port`	Sì	Il numero di porta su cui il server Redis ascolta le connessioni.	int	nullo
`username`	No	Il nome utente per l'autenti cazione.	stringa	null
`password`	No	La password per l'autenticazione.	stringa	Stringa vuota
`useSsl`	No	Speciifica se abilitare la SSL/ TLS crittografia per la connessio ne Redis.	booleano	false

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`database`	No	Il numero del database Redis da utilizzare. Redis supporta più database logici e questa proprietà specifica quale utilizzare.	int	0
`maxTotal`	No	Il numero massimo di connessio ni consentit e nel pool di connessioni Redis.	int	128
`maxIdle`	No	Il numero massimo di connessioni inattive consentit e nel pool di connessioni Redis.	int	128
`minIdle`	No	Il numero minimo di connessio ni inattive da mantenere nel pool di connessioni Redis.	int	16

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`testOnBorrow`	No	Un valore booleano che indica se convalidare le connessioni prima di prenderle in prestito dal pool.	booleano	true
`testOnReturn`	No	Un valore booleano che indica se convalidare le connessioni prima di restituirle al pool.	booleano	true
`testWhileIdle`	No	Un valore booleano che indica se convalidare periodicamente le connessioni inattive nel pool.	booleano	true
`testOnCreate`	No	Un valore booleano che indica se convalidare le connessioni quando vengono create.	booleano	true

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`minEvictableIdleTimeMillis`	No	Il periodo di tempo minimo (in millisecondi) in cui una connessione inattiva deve rimanere nel pool prima di poter essere rimossa.	Long	60.000 L
`timeBetweenEvictionRunsMillis`	No	Il tempo (in millisecondi) tra le esecuzioni successive del thread idle connection evictor.	Long	30.000 L
`numTestsPerEvictionRun`	No	Il numero massimo di connessioni da testare durante ogni esecuzione del thread idle connection evictor.	int	-1

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`blockWhenExhausted`	No	Un valore booleano che indica se bloccare e attendere che una connessione diventi disponibile quando il pool è esaurito.	booleano	true
`nettyThreads`	No	Il numero di thread Netty da utilizzare per gestire le connessioni Redis.	int	32
`subscriptionsPerConnection`	No	Il numero massimo di abbonamenti consentiti per connessione Redis.	int	10
`subscriptionConnectionPoolSize`	No	Il numero massimo di connessioni consentite nel pool di connessioni con abbonamento Redis.	int	100

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`pageSizeInBytes`	No	La dimensione predefinita della pagina in byte per le operazioni Redis.	Long	262144000
`readTimeout`	No	Il timeout di lettura in millisecondi per le operazioni Redis.	Long	2000
`timeToLiveMillis`	No	La durata (in millisecondi) per la quale una voce della cache rimane nella cache prima di essere considerata scaduta e rimossa. Se questa proprietà non è specificata, le voci della cache non scadranno automaticamente per impostazione predefinita.	Long	-1

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`useAsyncBatch`	No	Abilita l'esecuzione asincrona per le operazioni di scrittura in blocco di Redis per migliorare le prestazioni. Se impostato su false, torna alla modalità di esecuzione sincrona.	booleano	true
`useBatchInMemoryAtomic`	No	Abilita In-memory-atomic la modalità per le operazioni di lettura in batch di Redis. Se impostato su false, torna alla modalità batch in memoria predefinita.	booleano	false
`connectionPoolSize`	No	La dimensione del pool di connessioni utilizzato per le connessioni dei client Redisson al server Redis.	int	64

Nome proprietà	Obbligatorio?	Description	Valori	Predefinita
`connectionMinimumIdleSize`	No	Il numero minimo di connessioni inattive che Redisson manterrà nel suo pool di connessioni.	int	24
`idleConnectionTimeout`	No	Il timeout in millisecondi dopo il quale verrà chiusa una connessione inattiva nel pool.	int	10000
`connectTimeout`	No	Il timeout in millisecondi per stabilire una connessione al server Redis.	int	10000

Proprietà della cache Redis

Cache Redis Blusam

```
bluesam:
   cache: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                        # Optional
     username:                               # Optional
     password: ""                            # Optional
     useSsl: false                           # Optional
     database: 0                             # Optional
```

```
    maxTotal: 128                           # Optional
    maxIdle: 128                            # Optional
    minIdle: 16                             # Optional
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
    timeToLiveMillis: 60000                 # Optional
    connectionPoolSize: 64                  # Optional
    connectionMinimumIdleSize: 24           # Optional
    idleConnectionTimeout: 10000            # Optional
    connectTimeout: 10000                   # Optional
```

Cache Redis Blusam

```
bluesam:
   locks:
     cache: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
       hostName: localhost
       port: 6379
       mode: standalone                       # Optional
       username:                              # Optional
       password: ""                           # Optional
       useSsl: false                          # Optional
       database: 0                            # Optional
       maxTotal: 128                          # Optional
       maxIdle: 128                           # Optional
       minIdle: 16                            # Optional
       testOnBorrow: true                     # Optional
       testOnReturn: true                     # Optional
       testWhileIdle: true                    # Optional
       testOnCreate: true                     # Optional
```

```
      minEvictableIdleTimeMillis: 60000       # Optional
      timeBetweenEvictionRunsMillis: 30000    # Optional
      numTestsPerEvictionRun: -1              # Optional
      blockWhenExhausted: true                # Optional
      nettyThreads: 32                        # Optional
      subscriptionsPerConnection: 10          # Optional
      subscriptionConnectionPoolSize: 100     # Optional
      pageSizeInBytes: 8192                   # Optional
      readTimeout: 2000                       # Optional
```

Cache di sessione

```
spring:
  session:
    store-type: redis
# If the following redis properties are not specified gapwalk.redis configuration will
be used for this cache
jics:
  redis:
    hostName: localhost
    port: 6379
    mode: standalone                        # Optional
    username:                               # Optional
    password: ""                            # Optional
    useSsl: false                           # Optional
    database: 0                             # Optional
    maxTotal: 128                           # Optional
    maxIdle: 128                            # Optional
    minIdle: 16                             # Optional
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

Definizioni delle risorse JICS

```
jics:
   resource-definitions:
     store-type: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                        # Optional
     username:                               # Optional
     password: ""                            # Optional
     useSsl: false                           # Optional
     database: 0                             # Optional
     maxTotal: 128                           # Optional
     maxIdle: 128                            # Optional
     minIdle: 16                             # Optional
     testOnBorrow: true                      # Optional
     testOnReturn: true                      # Optional
     testWhileIdle: true                     # Optional
     testOnCreate: true                      # Optional
     minEvictableIdleTimeMillis: 60000       # Optional
     timeBetweenEvictionRunsMillis: 30000    # Optional
     numTestsPerEvictionRun: -1              # Optional
     blockWhenExhausted: true                # Optional
     nettyThreads: 32                        # Optional
     subscriptionsPerConnection: 10          # Optional
     subscriptionConnectionPoolSize: 100     # Optional
     pageSizeInBytes: 8192                   # Optional
     readTimeout: 2000                       # Optional
```

Code JICS TS

```
jics:
   parameters:
     tsqimpl: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   queues:
     ts:
       redis:
         hostName: localhost
```

```
        port: 6379
        mode: standalone                        # Optional
        username:                               # Optional
        password: ""                            # Optional
        useSsl: false                           # Optional
        database: 0                             # Optional
        maxTotal: 128                           # Optional
        maxIdle: 128                            # Optional
        minIdle: 16                             # Optional
        testOnBorrow: true                      # Optional
        testOnReturn: true                      # Optional
        testWhileIdle: true                     # Optional
        testOnCreate: true                      # Optional
        minEvictableIdleTimeMillis: 60000       # Optional
        timeBetweenEvictionRunsMillis: 30000    # Optional
        numTestsPerEvictionRun: -1              # Optional
        blockWhenExhausted: true                # Optional
        nettyThreads: 32                        # Optional
        subscriptionsPerConnection: 10          # Optional
        subscriptionConnectionPoolSize: 100     # Optional
        pageSizeInBytes: 8192                   # Optional
        readTimeout: 2000                       # Optional
```

Tracciatore della sessione

```
session-tracker:
   store-type: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                       # Optional
     username:                              # Optional
     password: ""                           # Optional
     useSsl: false                          # Optional
     database: 0                            # Optional
     maxTotal: 128                          # Optional
     maxIdle: 128                           # Optional
     minIdle: 16                            # Optional
     testOnBorrow: true                     # Optional
     testOnReturn: true                     # Optional
     testWhileIdle: true                    # Optional
```

```
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

Punto di controllo JCL

```
jcl:
   checkpoint:
     provider: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostname: localhost
     port: 6379
     mode: standalone                       # Optional
     username:                              # Optional
     password: ""                           # Optional
     useSsl: false                          # Optional
     database: 0                            # Optional
     maxTotal: 128                          # Optional
     maxIdle: 128                           # Optional
     minIdle: 16                            # Optional
     testOnBorrow: true                     # Optional
     testOnReturn: true                     # Optional
     testWhileIdle: true                    # Optional
     testOnCreate: true                     # Optional
     minEvictableIdleTimeMillis: 60000      # Optional
     timeBetweenEvictionRunsMillis: 30000   # Optional
     numTestsPerEvictionRun: -1             # Optional
     blockWhenExhausted: true               # Optional
     nettyThreads: 32                       # Optional
     subscriptionsPerConnection: 10         # Optional
     subscriptionConnectionPoolSize: 100    # Optional
     pageSizeInBytes: 8192                  # Optional
     readTimeout: 2000                      # Optional
```

Blocchi di file Gapwalk

```
filesLocks:
   enabled: true
   retryTime: 1000
   MaxRetry: 5
   provider: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                          # Optional
     username:                                 # Optional
     password: ""                              # Optional
     useSsl: false                             # Optional
     database: 0                               # Optional
     pool:
       maxTotal: 128                           # Optional
       maxIdle: 128                            # Optional
       minIdle: 16                             # Optional
       testOnBorrow: true                      # Optional
       testOnReturn: true                      # Optional
       testWhileIdle: true                     # Optional
       testOnCreate: true                      # Optional
       minEvictableIdleTimeMillis: 60000       # Optional
       timeBetweenEvictionRunsMillis: 30000    # Optional
       numTestsPerEvictionRun: -1              # Optional
       blockWhenExhausted: true                # Optional
       nettyThreads: 32                        # Optional
       subscriptionsPerConnection: 10          # Optional
       subscriptionConnectionPoolSize: 100     # Optional
       pageSizeInBytes: 8192                   # Optional
       readTimeout: 2000                       # Optional
```

Serrature Blu4iv

```
blu4iv.lock: redis
blu4iv.lock.timeout: 10 #(in millisecondes)
 # If the following redis properties are not specified gapwalk.redis configuration
will be used for this cache
blu4iv.lock.redis:
```

```
      hostName: localhost
      port: 6379
      mode: standalone                          # Optional
      username:                                 # Optional
      password: ""                              # Optional
      useSsl: false                             # Optional
      database: 0                               # Optional
      maxTotal: 128                             # Optional
      maxIdle: 128                              # Optional
      minIdle: 16                               # Optional
      testOnBorrow: true                        # Optional
      testOnReturn: true                        # Optional
      testWhileIdle: true                       # Optional
      testOnCreate: true                        # Optional
      minEvictableIdleTimeMillis: 60000         # Optional
      timeBetweenEvictionRunsMillis: 30000      # Optional
      numTestsPerEvictionRun: -1                # Optional
      blockWhenExhausted: true                  # Optional
      nettyThreads: 32                          # Optional
      subscriptionsPerConnection: 10            # Optional
      subscriptionConnectionPoolSize: 100       # Optional
      pageSizeInBytes: 8192                     # Optional
      readTimeout: 2000                         # Optional
```

Catalogo dei set di dati

```
datasimplifier:
   catalogImplementation: redis
   # If the following redis properties are not specified gapwalk.redis configuration
 will be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                         # Optional
     username:                                # Optional
     password: ""                             # Optional
     useSsl: false                            # Optional
     database: 0                              # Optional
     maxTotal: 128                            # Optional
     maxIdle: 128                             # Optional
     minIdle: 16                              # Optional
     testOnBorrow: true                       # Optional
     testOnReturn: true                       # Optional
```

```
    testWhileIdle: true                    # Optional
    testOnCreate: true                     # Optional
    minEvictableIdleTimeMillis: 60000      # Optional
    timeBetweenEvictionRunsMillis: 30000   # Optional
    numTestsPerEvictionRun: -1             # Optional
    blockWhenExhausted: true               # Optional
    nettyThreads: 32                       # Optional
    subscriptionsPerConnection: 10         # Optional
    subscriptionConnectionPoolSize: 100    # Optional
    pageSizeInBytes: 8192                  # Optional
    readTimeout: 2000                      # Optional
```

Gestore segreto per le cache Redis

Il `application-main.yaml` file può fare riferimento all'ARN segreto per le cache Redis. Per informazioni su come effettuare l'integrazione per recuperare in modo sicuro Gestione dei segreti AWS i dettagli della connessione Redis in fase di esecuzione, consulta. the section called "AWS I segreti di Transform for mainframe Runtime"

Configurare la sicurezza per le applicazioni Gapwalk

I seguenti argomenti descrivono come proteggere le applicazioni Gapwalk.

È responsabilità dell'utente fornire la configurazione corretta per garantire che l'uso del framework AWS Transform for mainframe sia sicuro.

Tutte le funzionalità relative alla sicurezza sono disattivate per impostazione predefinita.
Per abilitare l'autenticazione (e CSRF, XSS, CSP e così via), imposta su e su. `gapwalk-application.security` `enabled` `gapwalk-application.security.identity` `oauth`

Argomenti

- Configurare l'accessibilità degli URI per le applicazioni Gapwalk
- Configurare l'autenticazione per le applicazioni Gapwalk
- Configurare la limitazione della velocità per AWS Transform for mainframe Runtime

Configurare l'accessibilità degli URI per le applicazioni Gapwalk

Questo argomento descrive come configurare il filtraggio delle URIs applicazioni Gapwalk. Questa funzionalità non richiede un provider di identità (IdP).

Per bloccare un elenco di URIs, aggiungi le due righe seguenti all'`application-main.yml`applicazione modernizzata, sostituendole *URI-1URI-2*, e così via, con quella URIs che desideri bloccare.

```
gapwalk-application.security.filterURIs: enabled
gapwalk-application.security.blockedURIs: URI-1, URI-2, URI-3
```

Configurare l'autenticazione per le applicazioni Gapwalk

Per configurare OAuth2 l'autenticazione per l'applicazione Gapwalk, è necessario configurare un provider di identità (IdP) e integrarlo con l'applicazione. Questa guida illustra i passaggi per utilizzare Amazon Cognito o Keycloak come IdP. Con Amazon Cognito, puoi aggiornare il file di configurazione dell'applicazione con i dettagli del pool di utenti Cognito. Con Keycloak, puoi controllare l'accesso alle tue applicazioni APIs e alle tue risorse in base ai ruoli assegnati all'utente.

Argomenti

- Configura l' OAuth2 autenticazione Gapwalk con Amazon Cognito
- Configura l'autenticazione OAuth2 Gapwalk con Keycloak

Configura l' OAuth2 autenticazione Gapwalk con Amazon Cognito

Questo argomento descrive come configurare OAuth2 l'autenticazione per le applicazioni Gapwalk utilizzando Amazon Cognito come provider di identità (IdP).

Prerequisiti

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 di Gapwalk. Per ulteriori informazioni, consulta la Guida per sviluppatori di Amazon Cognito .

Le informazioni: ClientRegistration

client-id

L'ID del ClientRegistration. Nel nostro esempio lo sarà. PlanetsDemo

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 dell'endpoint del token per il server di autorizzazione.

endpoint jwks

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

URI di reindirizzamento

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

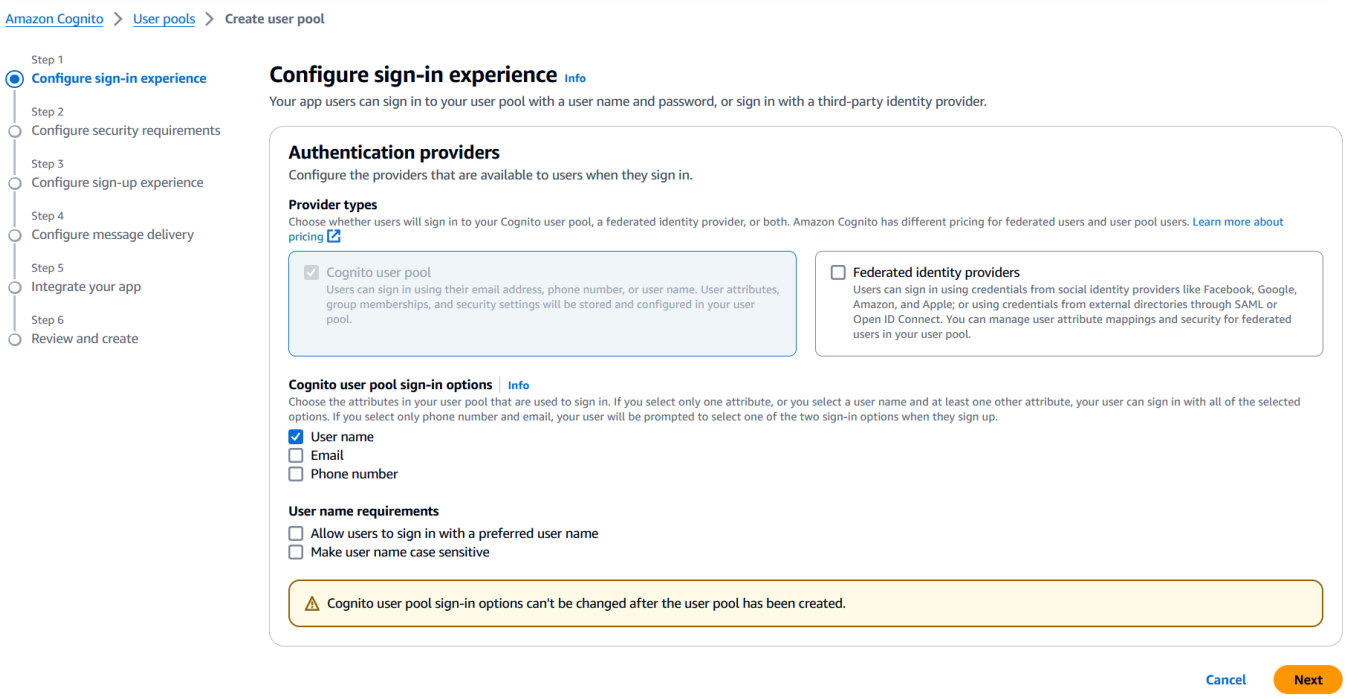
Configurazione di Amazon Cognito

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.

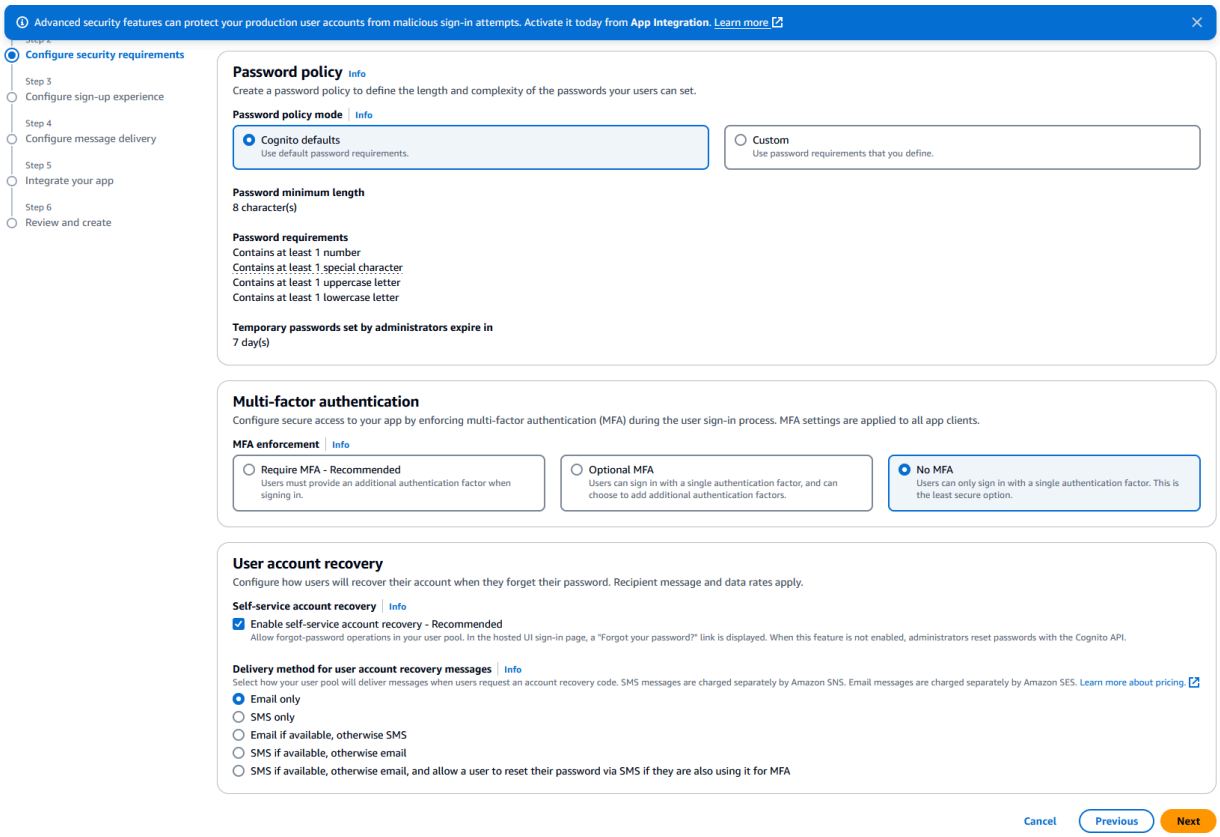
> **Note**
>
> Se utilizzi un altro IdP, puoi saltare questo passaggio.

Crea un pool di utenti

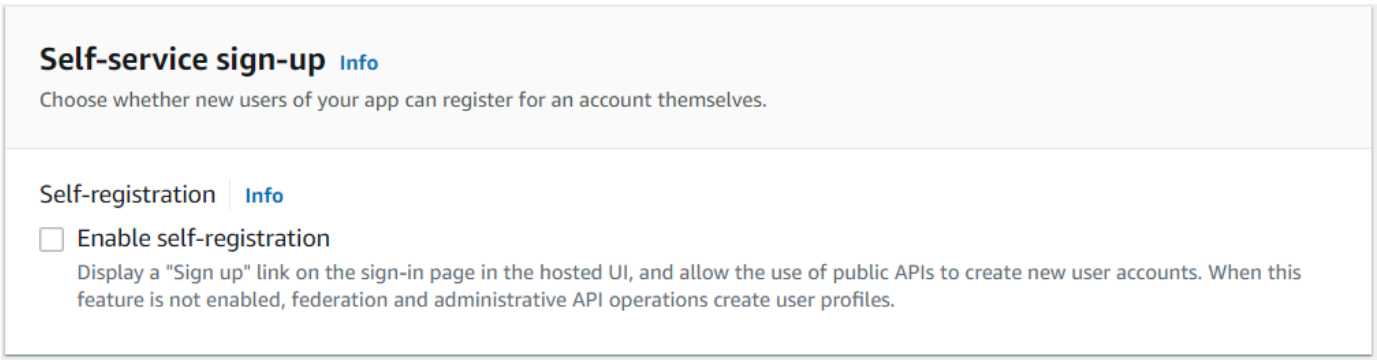
1. Accedi ad Amazon Cognito in Console di gestione AWS ed esegui l'autenticazione utilizzando le tue credenziali. AWS
2. Scegli User Pools (Pool di utenti).
3. Scegli Create a User Pool (Crea un bacino d'utenza).
4. In Configura l'esperienza di accesso, mantieni il tipo di provider predefinito del pool 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.



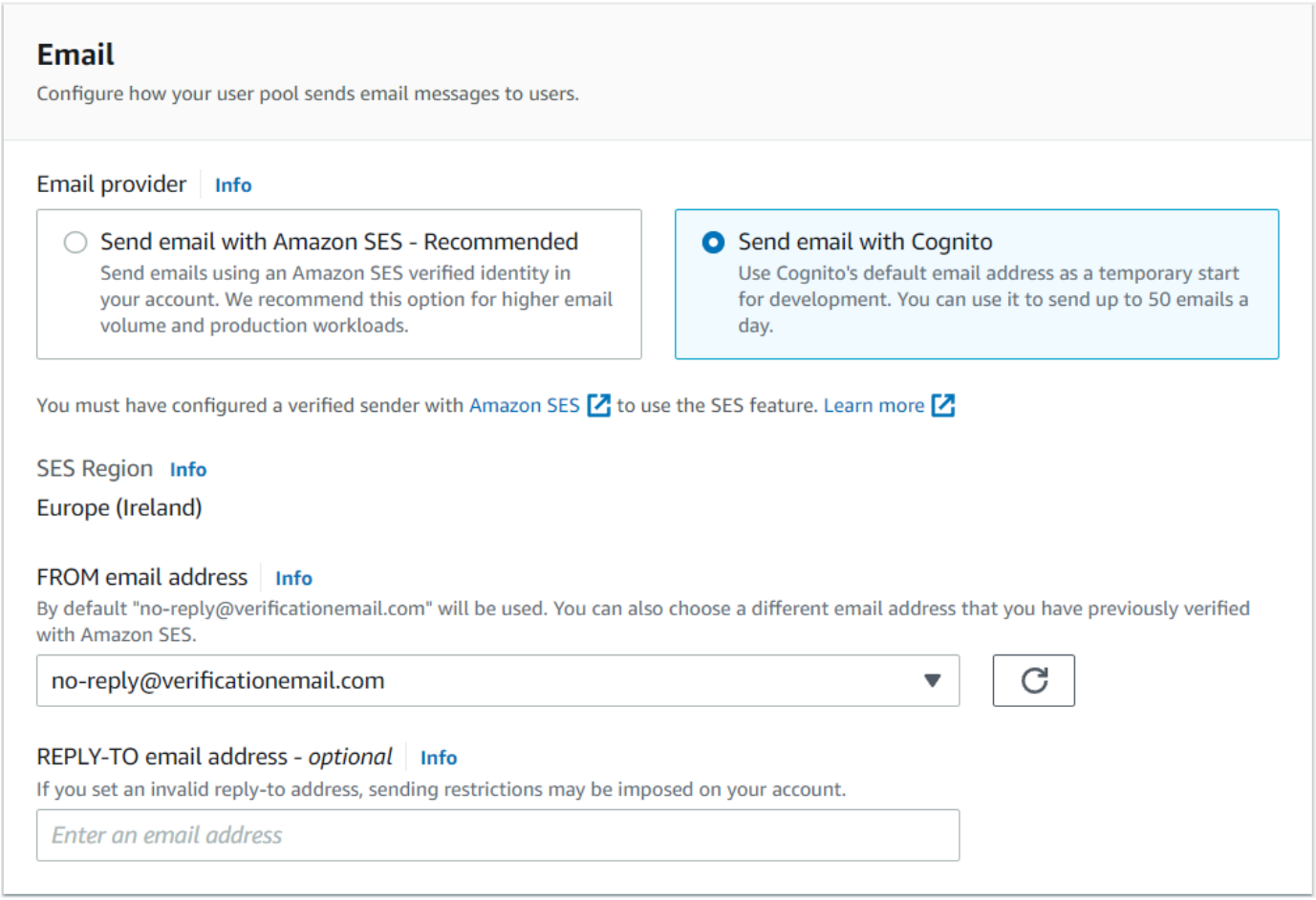
5. In Configura i requisiti di sicurezza, mantieni le impostazioni predefinite e disabilita l'autenticazione a più fattori scegliendo Nessuna MFA, quindi scegli Avanti.



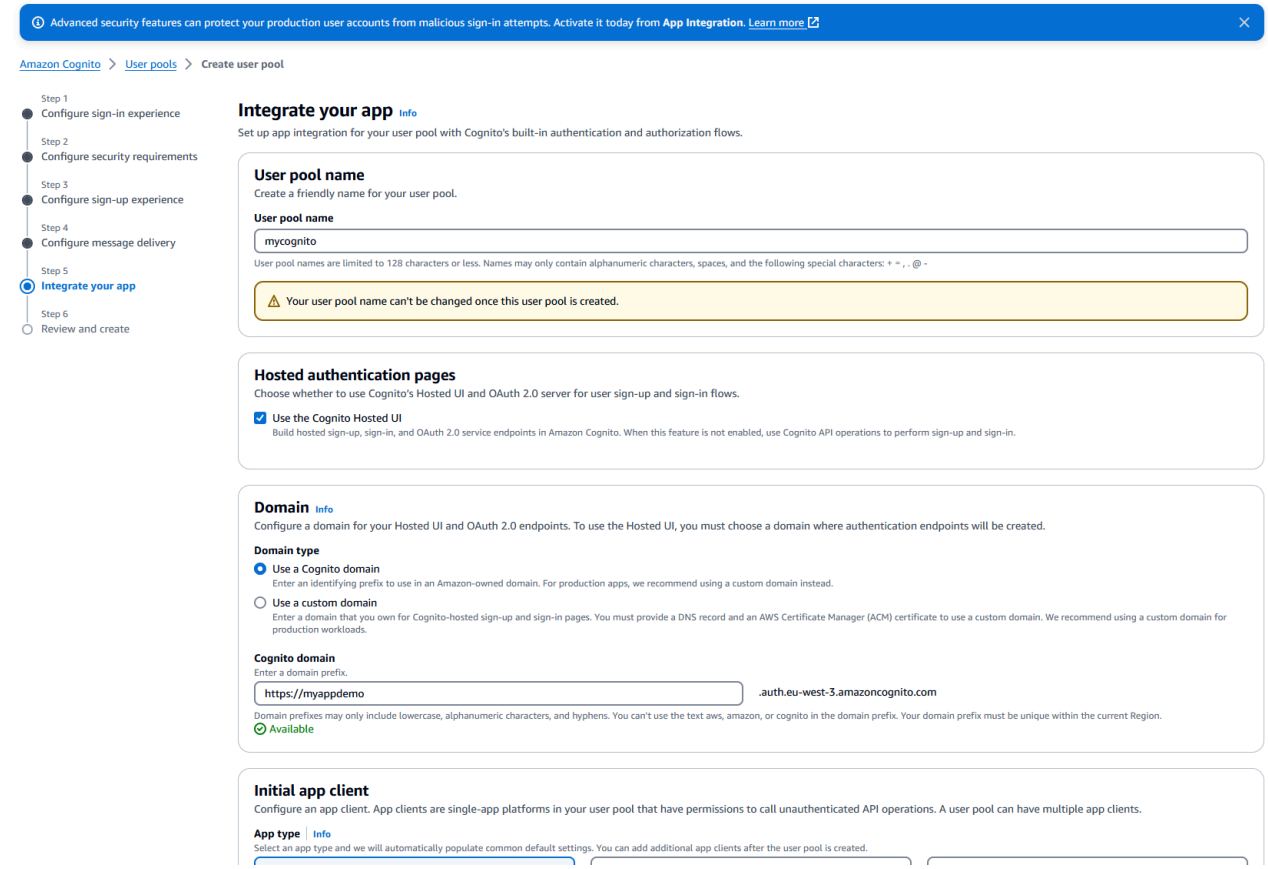
6. Come misura di sicurezza, disattiva Abilita l'autoregistrazione, quindi scegli Avanti.



7. Scegli Invia email con Cognito, quindi scegli Avanti.



8. In Integra la tua app, specifica un nome per il tuo pool di utenti. Nelle pagine di autenticazione ospitate, scegli Usa l'interfaccia utente ospitata da Cognito.



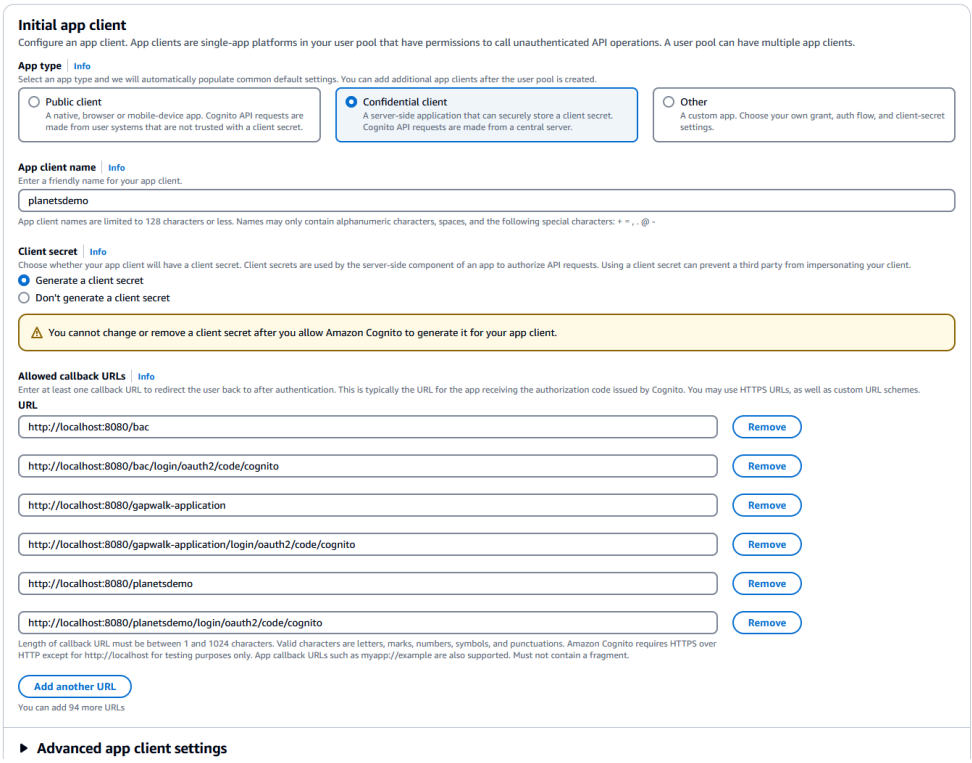
9. Per semplicità, in Dominio, scegli Usa un dominio Cognito e inserisci un prefisso di dominio, ad esempio. `https://planetsdemo` L'app demo deve essere aggiunta come client.

   a. Nel client iniziale dell'app, scegli Client riservato. Inserisci il nome del client dell'app, ad esempio**`planetsdemo`**, quindi scegli Genera un segreto client.

   b. In URL di callback consentito, inserisci l'URL a cui reindirizzare l'utente dopo l'autenticazione. L'URL deve terminare con. `/login/oauth2/code/cognito` Ad esempio, per la nostra applicazione e le applicazioni di backend Gapwalk e BAC:

```
http://localhost:8080/bac
      http://localhost:8080/bac/login/oauth2/code/cognito
      http://localhost:8080/gapwalk-application
      http://localhost:8080/gapwalk-application/login/oauth2/code/cognito
      http://localhost:8080/planetsdemo
      http://localhost:8080/planetsdemo/login/oauth2/code/cognito
```

   Puoi modificare l'URL in un secondo momento.



c. In Connessione consentita, URLs inserisci l'URL della pagina di disconnessione a cui desideri reindirizzare Amazon Cognito quando l'applicazione disconnette gli utenti. Ad esempio, per le applicazioni backend Gapwalk e BAC:

```
http://localhost:8080/bac/logout
http://localhost:8080/gapwalk-application/logout
http://localhost:8080/planetsdemo/logout
```

Puoi modificare l'URL in un secondo momento.

d. Mantieni i valori predefiniti nelle sezioni Impostazioni avanzate del client dell'app e Attribuisci autorizzazioni di lettura e scrittura.

e. Scegli Next (Successivo).

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. Console di gestione AWS 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).

Creazione del ruolo

Nella scheda Gruppi, crea 3 gruppi (SUPER_ADMIN, ADMIN e USER) e associa il tuo utente a uno o più di questi gruppi. Questi ruoli vengono successivamente mappati su ROLE_SUPER_ADMIN, ROLE_ADMIN e ROLE_USER dall'applicazione Gapwalk per consentire l'accesso ad alcune chiamate REST API con restrizioni.

L'applicazione scope-to-role implementa una mappatura OAuth2 gerarchica che funziona con più provider di identità. Quando i token JWT emessi da Cognito vengono utilizzati per l'autorizzazione del server di risorse, gli ambiti definiti nel token vengono mappati automaticamente ai ruoli corrispondenti.

Integra Amazon Cognito nell'applicazione Gapwalk

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

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

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: "<redirect-uri>"
        provider:
          cognito:
            issuer-uri: ${gapwalk-application.security.issuerUri}
```

```
            authorization-uri: ${gapwalk-application.security.domainName}/oauth2/
authorize
            jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/
jwks.json
            token-uri: ${gapwalk-application.security.domainName}/oauth2/token
            user-name-attribute: username
      resourceserver:
        jwt:
          jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/jwks.json
```

Sostituisci i seguenti segnaposto come descritto:

1. Accedi ad Amazon Cognito in Console di gestione AWS 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 ID ***pool-id*** del pool di utenti.
3. Scegli l'integrazione delle app dove puoi trovare la tua*your-cognito-domain*, quindi vai su App client e analisi e scegli la tua app.
4. Nel client App: YourApp puoi trovare *client-nameclient-id*, e *client-secret* (Mostra il segreto del client).
5. *region-id*corrisponde all'ID della AWS regione in cui hai creato l'utente e il pool di utenti di Amazon Cognito. Esempio: `eu-west-3`.
6. Per *redirect-uri* inserisci l'URI che hai specificato per Allowed callback URL. Nel nostro esempio lo è`http://localhost:8080/planetsdemo/login/oauth2/code/cognito`.

Ora puoi distribuire la tua applicazione Gapwalk e utilizzare l'utente creato in precedenza per accedere alla tua app.

Configura l'autenticazione OAuth2 Gapwalk con Keycloak

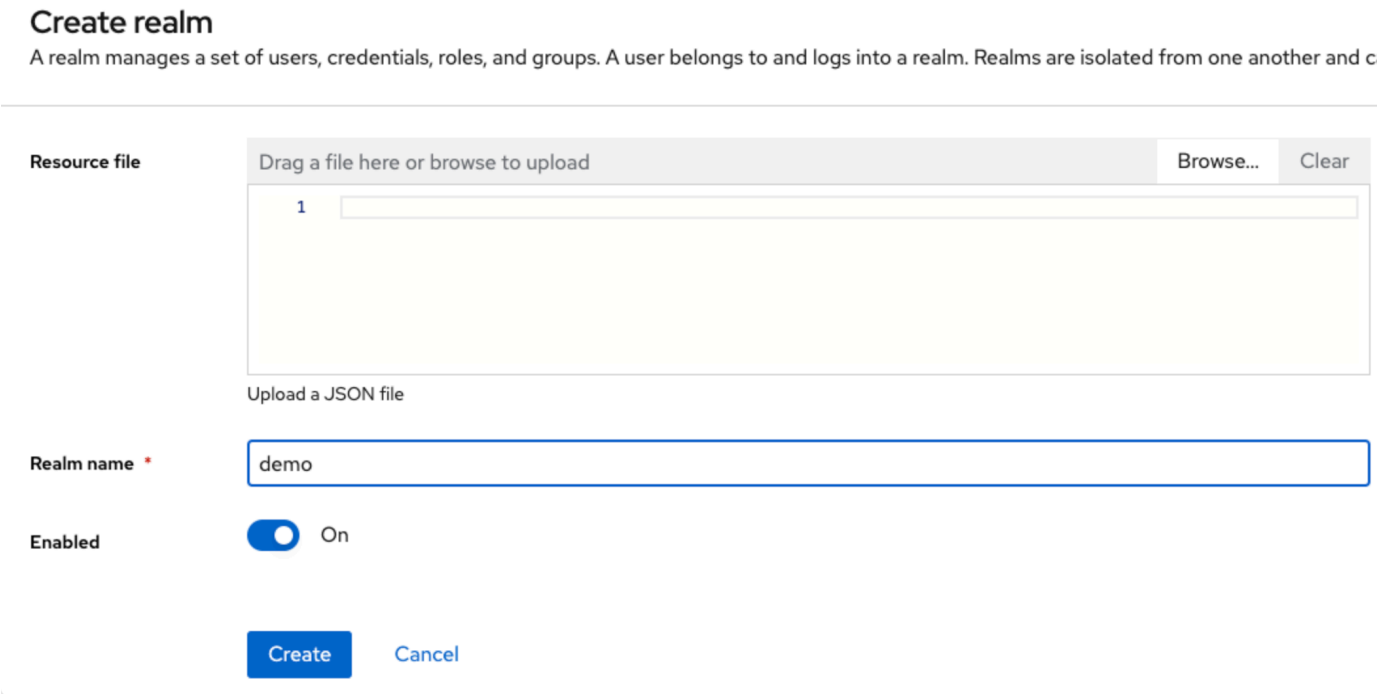
Questo argomento descrive come configurare OAuth2 l'autenticazione per le applicazioni Gapwalk utilizzando Keycloak come provider di identità (IdP). In questo tutorial utilizziamo Keycloak 24.0.0.

Prerequisiti

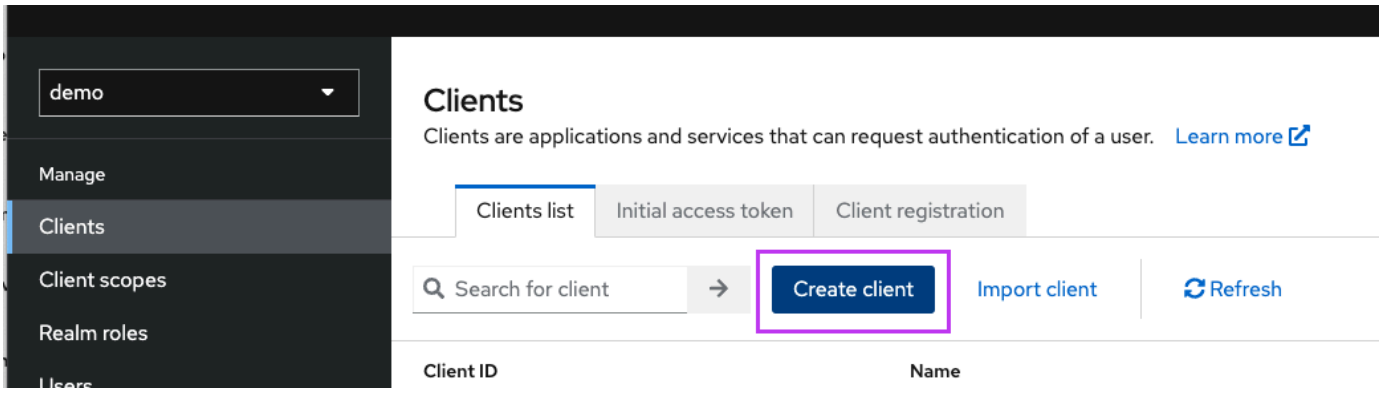
- Keycloak
- Applicazione Gapwalk

Configurazione Keycloak

1. Vai alla dashboard di Keycloak nel tuo browser web. Le credenziali predefinite sono admin/admin. Vai alla barra di navigazione in alto a sinistra e crea un realm con il nome**`demo`**, come mostrato nell'immagine seguente.



2. Crea un client con il nome**`app-demo`**.



`localhost:8080`Sostituiscilo con l'indirizzo della tua applicazione Gapwalk

General settings

Client ID * app-demo

Name

Description

Always display in UI Off

Access settings

Root URL http://localhost:8080

Home URL

Valid redirect URIs
- http://localhost:8080/*
- https://localhost:8080/*

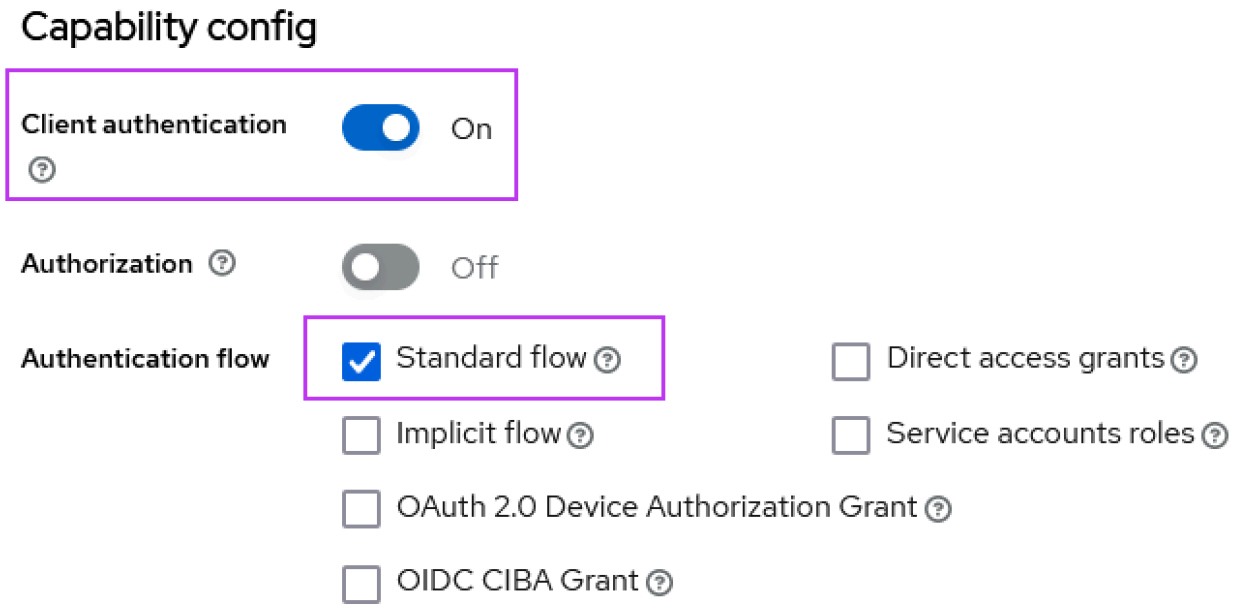
Add valid redirect URIs

Valid post logout redirect URIs
- http://localhost:8080/*
- https://localhost:8080/*

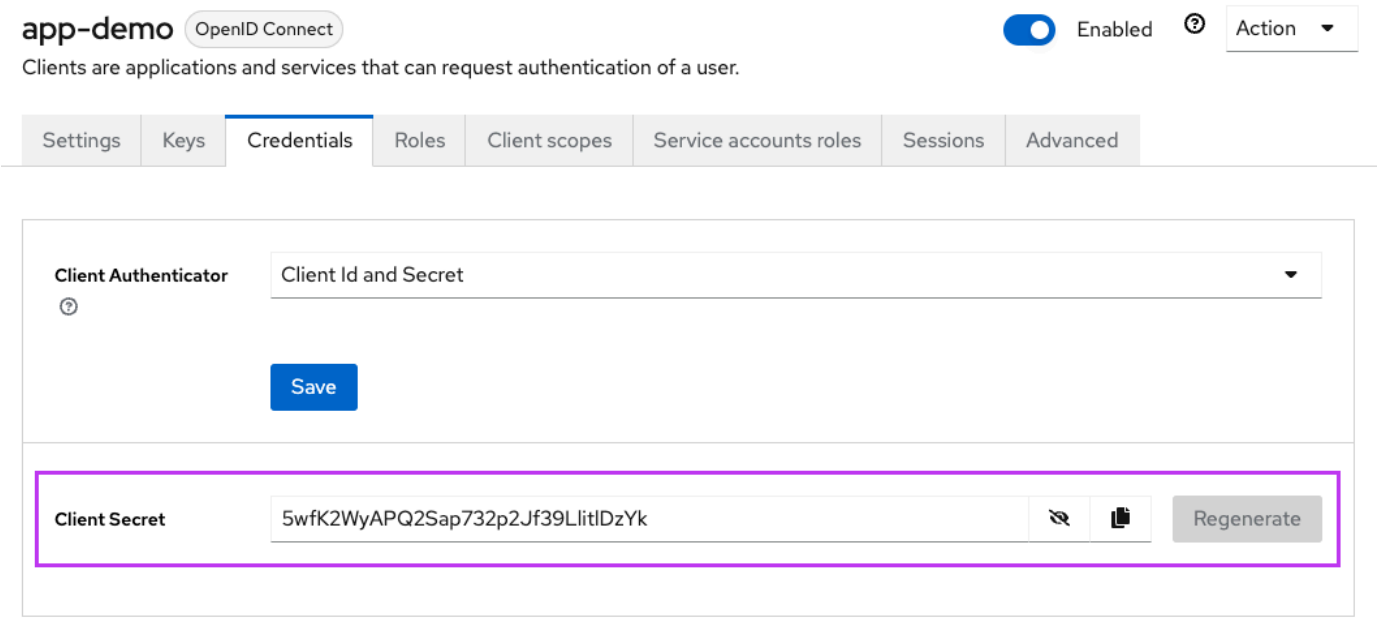
Add valid post logout redirect URIs

Web origins
- +

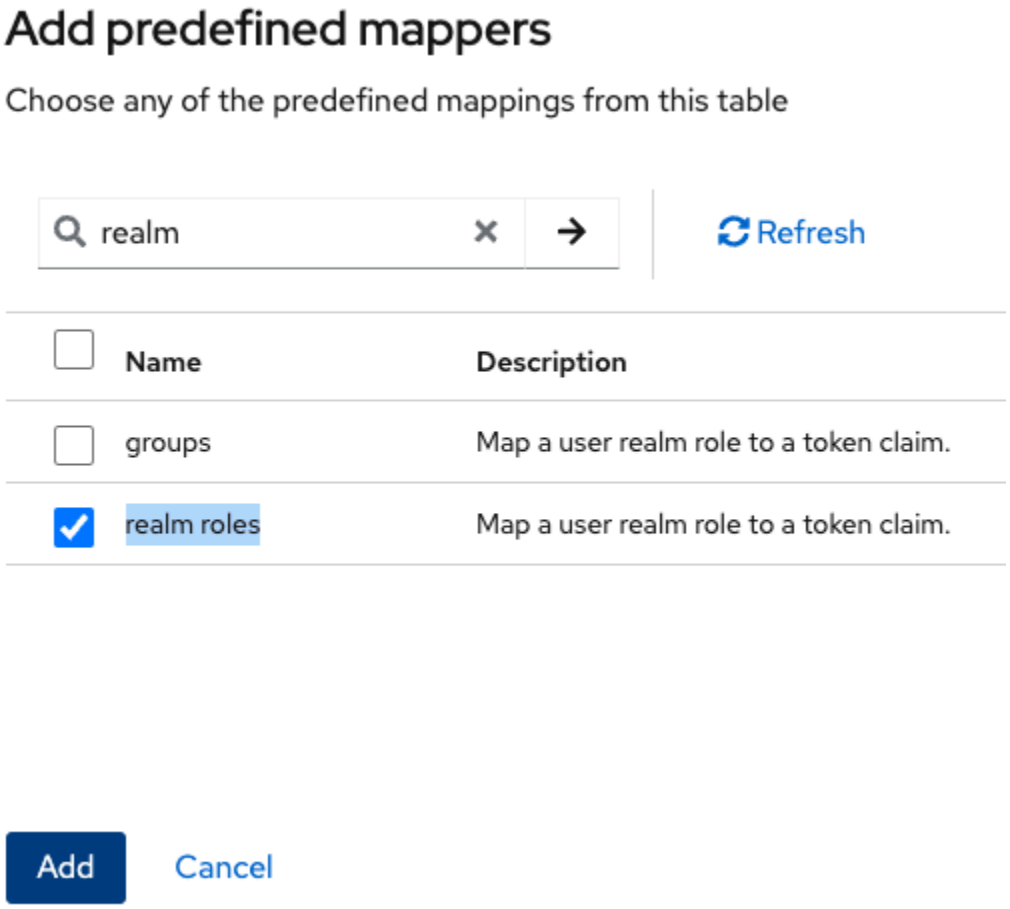
Add web origins



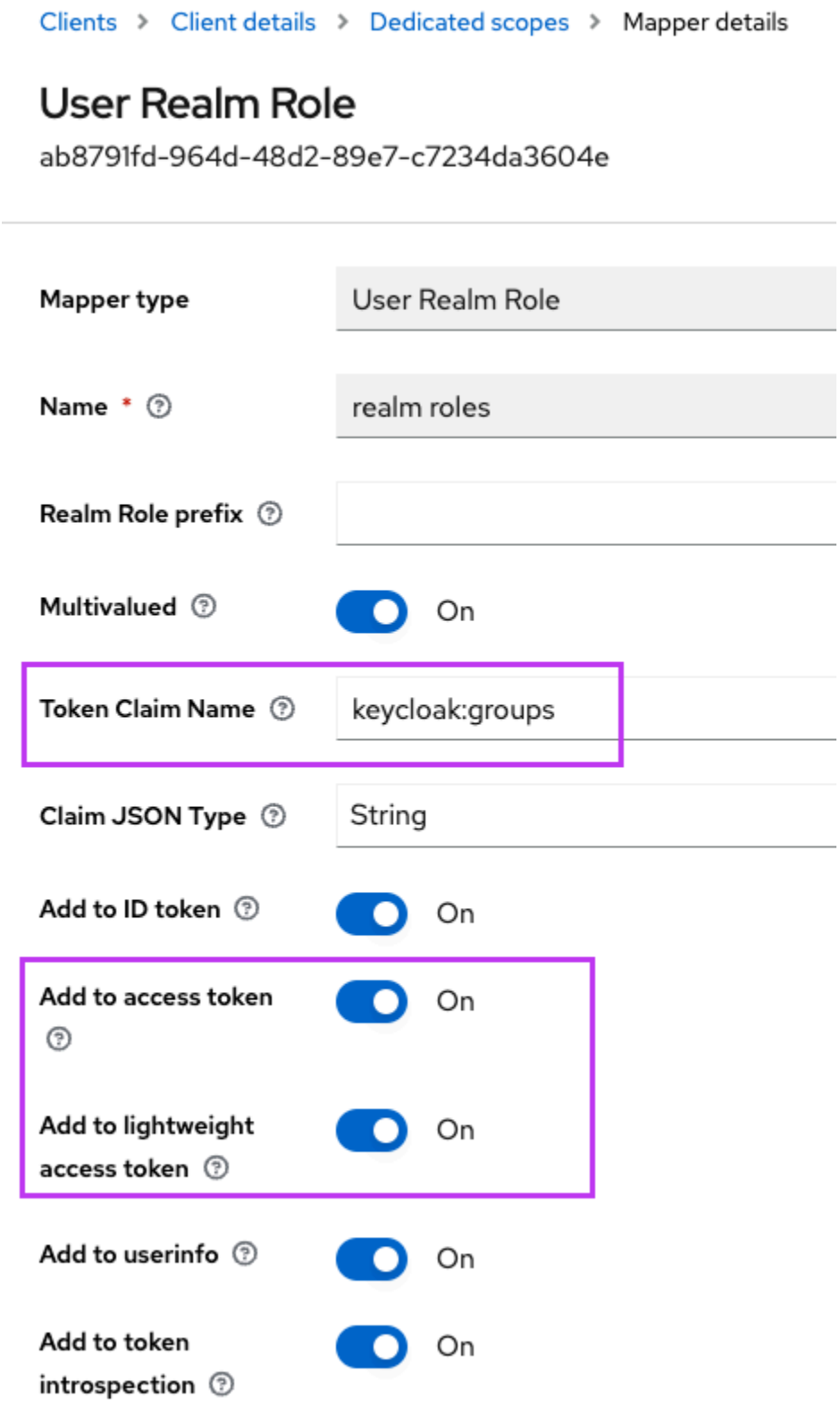
3. Per rendere segreto il tuo client, scegli Clients, poi app-demo, quindi Credenziali.



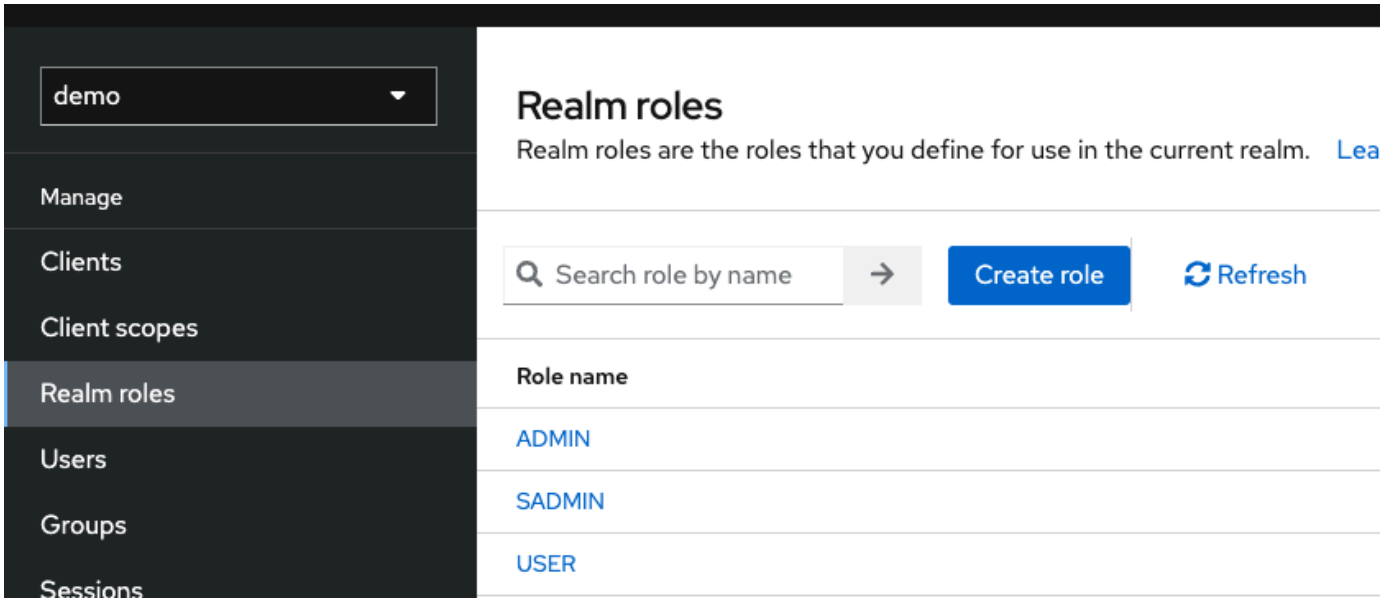
4. Scegli Clienti, quindi Ambiti client, quindi Aggiungi mappatore predefinito. Scegli i ruoli del realm.



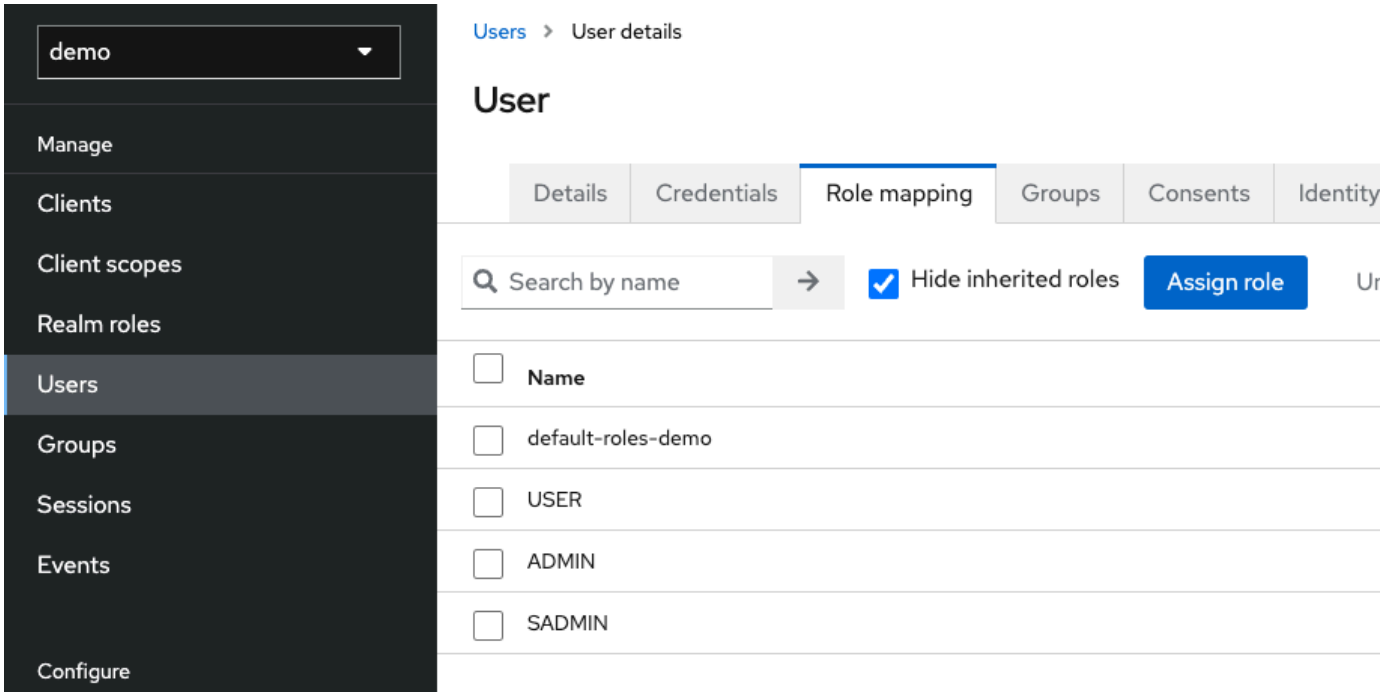
5. Modifica il tuo ruolo di realm con la configurazione mostrata nell'immagine seguente.


Clients > Client details > Dedicated scopes > Mapper details
User Realm Role
ab8791fd-964d-48d2-89e7-c7234da3604e
Mapper type
User Realm Role
Name *
realm roles
Realm Role prefix
Multivalued
On
Token Claim Name
keycloak:groups
Claim JSON Type
String
Add to ID token
On
Add to access token
On
Add to lightweight access token
On
Add to userinfo
On
Add to token introspection
On

6. Ricorda il Token Claim Name definito. Avrai bisogno di questo valore nella definizione delle impostazioni Gapwalk per la `gapwalk-application.security.claimGroupName` proprietà.



7. Scegli i ruoli di Realms e crea 3 ruoli:**`SUPER_ADMIN`**, **`ADMIN`** e. **`USER`** Questi ruoli vengono successivamente mappati su `ROLE_SUPER_ADMIN` e `ROLE_USER` dall'applicazione Gapwalk per poter accedere ad alcune chiamate API REST con restrizioni. `ROLE_ADMIN`



Integra Keycloak nell'applicazione Gapwalk

Modifica il tuo come segue: `application-main.yml`

```
gapwalk-application.security: enabled
gapwalk-application.security.identity: oauth
gapwalk-application.security.issuerUri: http://<KEYCLOAK_SERVER_HOSTNAME>/realms/
<YOUR_REALM_NAME>
gapwalk-application.security.claimGroupName: "keycloak:groups"

gapwalk-application.security.userAttributeName: "preferred_username"
# Use "username" for cognito,
#     "preferred_username" for keycloak
#      or any other string

spring:
  security:
    oauth2:
      client:
        registration:
          demo:
            client-id: <YOUR_CLIENT_ID>
            client-name: Demo App
            client-secret: <YOUR_CLIENT_SECRET>
            provider: keycloak
            authorization-grant-type: authorization_code
            scope: openid
            redirect-uri: "{baseUrl}/login/oauth2/code/{registrationId}"
        provider:
          keycloak:
            issuer-uri: ${gapwalk-application.security.issuerUri}
            authorization-uri: ${gapwalk-application.security.issuerUri}/protocol/
openid-connect/auth
            jwk-set-uri: ${gapwalk-application.security.issuerUri}/protocol/openid-
connect/certs
            token-uri: ${gapwalk-application.security.issuerUri}/protocol/openid-
connect/token
            user-name-attribute: ${gapwalk-application.security.userAttributeName}
      resourceserver:
        jwt:
          jwk-set-uri: ${gapwalk-application.security.issuerUri}/protocol/openid-
connect/certs
```

Sostituisci *<KEYCLOAK_SERVER_HOSTNAME><YOUR_REALM_NAME>*,*<YOUR_CLIENT_ID>*, e *<YOUR_CLIENT_SECRET>* con il nome host del server Keycloak, il nome dell'area, l'ID cliente e il segreto del cliente.

Configurare la limitazione della velocità per AWS Transform for mainframe Runtime

AWS Transform for mainframe Runtime include funzionalità integrate di limitazione della velocità per proteggere l'applicazione gapwalk da richieste eccessive e potenziali abusi. Il sistema di limitazione della velocità utilizza l'algoritmo Token Bucket per fornire sia una capacità di burst che una limitazione sostenuta della velocità.

Argomenti

- Panoramica sulla limitazione della velocità
- Proprietà della configurazione
- Abilita la limitazione della velocità
- Identificazione del cliente
- Intestazioni relative ai limiti di velocità
- Gestione della memoria

Panoramica sulla limitazione della velocità

Il sistema di limitazione della velocità offre le seguenti funzionalità:

Algoritmo Token Bucket

- Consente il traffico burst fino alla capacità di burst configurata
- Ricarica i token a una velocità costante in base alle richieste al minuto
- Fornisce una limitazione fluida della velocità senza bloccare i picchi di traffico legittimi

Identificazione del cliente

- Identifica i client in base all'indirizzo IP con supporto proxy
- Supporti X-Forwarded-For e X-Real-IP intestazioni
- Gestisce scenari di bilanciamento del carico e proxy inverso

Gestione automatica della memoria

- Pulisce automaticamente i periodi con limiti di velocità scaduti
- Intervalli di pulizia e tempi di scadenza configurabili

- Previene le perdite di memoria nelle applicazioni a esecuzione prolungata

Integrazione HTTP

- Restituisce HTTP 429 (Troppe richieste) quando i limiti vengono superati
- Include le intestazioni dei limiti di velocità standard nelle risposte
- Fornisce ai clienti informazioni da riprovare

Proprietà della configurazione

Configura la limitazione della velocità nel tuo file: `application-main.yaml`

```
gapwalk:
  ratelimiting:
    enabled: true                                    # Enable/disable rate limiting
    requestsPerMinute: 1000                          # Sustained rate limit per
 minute
    burstCapacity: 1500                              # Maximum burst requests
 allowed
    includeHeaders: true                             # Include X-RateLimit-*
 headers
    cleanupIntervalMinutes: 5                        # Cleanup interval for expired
 buckets
    bucketExpiryHours: 1                             # Hours after which unused
 buckets expire
    errorMessage: "Too many requests. Try again later."  # Custom error message
    whitelistIps: ""                                 # Comma-separated IPs to
 bypass limiting
    perEndpointLimiting: false                       # Apply limits per endpoint
 (not implemented)
```

Descrizioni delle proprietà

abilitato

Switch principale per abilitare o disabilitare la funzionalità di limitazione della velocità. Impostazione predefinita: `false`

requestsPerMinute

Numero di richieste consentite al minuto per la limitazione sostenuta della velocità. Rappresenta la frequenza di ricarica del token. Impostazione predefinita: `1000`

Burst Capacity

Numero massimo di richieste consentite in un burst prima dell'applicazione del limite di velocità. Dovrebbe essere superiore `requestsPerMinute` a quello necessario per consentire picchi di traffico. Impostazione predefinita: `1500`

Includi le intestazioni

Se includere le intestazioni dei limiti di velocità standard (`X-RateLimit-Limit`,`X-RateLimit-Remaining`,`X-RateLimit-Reset`) nelle risposte HTTP. Impostazione predefinita: `true`

cleanupIntervalMinutes

Intervallo in minuti tra la pulizia automatica dei periodi limite di velocità scaduti. Aiuta a prevenire perdite di memoria. Impostazione predefinita: `5`

bucketExpiryHours

Tempo in ore dopo il quale i periodi limite di velocità non utilizzati vengono considerati scaduti e idonei alla pulizia. Impostazione predefinita: `1`

errorMessage

Messaggio di errore personalizzato restituito nella risposta JSON quando viene superato il limite di velocità. Impostazione predefinita: `"Too many requests. Try again later."`

Suggerimenti sulla lista bianca

Elenco separato da virgole di indirizzi IP che aggirano completamente la limitazione della velocità. Utile per controlli sanitari o sistemi affidabili. Impostazione predefinita: `empty`

perEndpointLimiting

Se applicare limiti di velocità separati per endpoint anziché solo per client. Attualmente non implementato. Impostazione predefinita: `false`

Abilita la limitazione della velocità

Per abilitare la limitazione della velocità con le impostazioni predefinite:

```
gapwalk:
  ratelimiting:
    enabled: true
```

Identificazione del cliente

Il sistema di limitazione della velocità identifica i client utilizzando il seguente ordine di priorità:

1. X-Forwarded-For intestazione (primo IP se separato da virgole)
2. X-Real-IP intestazione
3. Indirizzo remoto dalla richiesta HTTP

Ciò garantisce una corretta identificazione del client quando l'applicazione è responsabile:

- Sistemi di load balancer
- Proxy inversi
- CDNs
- Gateway API

Esempio di identificazione del cliente

```
# Direct connection
Client IP: 192.168.1.100

# Behind load balancer with X-Forwarded-For
X-Forwarded-For: 203.0.113.45, 192.168.1.100
Client IP: 203.0.113.45 (first IP used)

# Behind reverse proxy with X-Real-IP
X-Real-IP: 203.0.113.45
Client IP: 203.0.113.45
```

Intestazioni relative ai limiti di velocità

Quando `includeHeaders` è abilitata, le seguenti intestazioni vengono aggiunte alle risposte HTTP:

X- RateLimit -Limite

Il limite di velocità per il cliente (richieste al minuto)

X- RateLimit -Rimanente

Il numero di richieste rimanenti nella finestra del limite di velocità corrente

X- RateLimit -Ripristina

L'ora in cui viene ripristinata la finestra del limite di velocità (timestamp Unix)

Esempi di intestazioni di risposta

```
X-RateLimit-Limit: 1000
X-RateLimit-Remaining: 847
X-RateLimit-Reset: 1640995200
```

Il limite di frequenza ha superato la risposta

Quando viene superato il limite di velocità, il sistema restituisce:

Stato HTTP

429 Troppe richieste

Content-Type

application/json

Riprova dopo

Numero di secondi di attesa prima di riprovare

```
{
  "error": "Rate limit exceeded",
  "message": "Too many requests. Try again later.",
  "retryAfter": 60,
  "timestamp": 1640995140000
}
```

Gestione della memoria

Il sistema di limitazione della velocità gestisce automaticamente la memoria per prevenire perdite nelle applicazioni con esecuzione prolungata:

Pulizia automatica

- Funziona ogni minuto `cleanupIntervalMinutes`
- Rimuove i secchi inutilizzati da ore `bucketExpiryHours`

- Registra l'attività di pulizia per il monitoraggio

Efficienza della memoria

- Utilizza strutture di dati simultanee per la sicurezza dei thread
- Creazione di Lazy Bucket (solo quando necessario)
- Implementazione efficiente del token bucket

Monitoraggio dell'attività di pulizia

Controlla i registri per i messaggi di pulizia:

```
INFO  RateLimitingService - Cleaned up 15 expired rate limiting buckets
```

AWS Codici di errore di runtime Transform for mainframe

Durante il ciclo di vita di un'applicazione che utilizza AWS Transform for mainframe Runtime, possono verificarsi errori per vari motivi, ad esempio errori di configurazione o problemi di rete. I codici di errore gestiti da AWS Transform for mainframe Runtime sono documentati qui con la causa e le azioni per risolverli. I codici di errore gestiti da AWS Transform for mainframe Runtime seguono questa convenzione:`BA-X1234`, dove X è una lettera e 1234 sono quattro cifre. In questa documentazione, gli errori sono classificati per prima lettera e poi per cifre.

Argomenti

- AWS Codici di errore generici di Transform for mainframe Runtime
- AWS Codici di errore di runtime Transform for mainframe relativi a Blusam
- AWS Codici di errore di runtime Transform for mainframe relativi ai batch
- AWS Codici di errore Transform for mainframe Runtime relativi a Datasimplifier
- AWS Transform for mainframe Runtime Codici di errore relativi ai file
- AWS Codici di errore di runtime Transform for mainframe relativi a CL Command Programs
- AWS Codici di errore di runtime Transform for mainframe relativi a JICS
- AWS Codici di errore di runtime Transform for mainframe relativi a ADABAS
- AWS Codici di errore di runtime Transform for mainframe relativi alla coda
- AWS Codici di errore Redis Transform for mainframe Runtime
- AWS Codici di errore di runtime Transform for mainframe relativi ai programmi di utilità

AWS Codici di errore generici di Transform for mainframe Runtime

Codici di errore generici, con il prefisso. BA-A

Errori di configurazione della sicurezza

Impostazioni di sicurezza globali

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-A0001	Fatale	Valore non valido per. gapwalk-application.security Solo enabled o disabled sono valori supportati. Se non è impostata, la sicurezza è disabilitata per impostazione predefinita. Imposta gapwalk-application.security: enabled per attivare le funzionalità di sicurezza globali (autenticazione XSS, CORS, CSRF, OAUTH) o per disattivarle. gapwalk-application.security: disabled Rimuovi la proprietà per utilizzare il valore predefinito (). disabled	

Configurazione dell'identità

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-A0011	Warn	La proprietà non gapwalk-application.security.identity deve essere impostata quando la protezione è disabilitata. La configurazione corrente non è coerente. Abilita la sicurezza gapwalk-application.security: enabled impostando o rimuovendo la	

Chiave	Gravità	Testo	Dettagli aggiuntivi
		`gapwalk-application.security.identity` proprietà. La configurazione dell'identità è applicabile solo quando la sicurezza è abilitata.	
`BA-A0012`	Fatale	Proprietà `gapwalk-application.security.identity` richiesta mancante. Questa proprietà è obbligatoria quando la sicurezza è abilitata. Imposta `gapwalk-application.security.identity: oauth` per utilizzare OAuth l'autenticazione (consigliata) o `gapwalk-application.security.identity: json` per utilizzare l'autenticazione JSON quando la sicurezza è abilitata.	
`BA-A0013`	Fatale	Valore non valido per. `gapwalk-application.security.identity` Solo `json` o `oauth` sono valori supportati. OAuth è il metodo di autenticazione consigliato. Impostato `gapwalk-application.security.identity: oauth` per utilizzare OAuth l'autenticazione (consigliato) o `gapwalk-application.security.identity: json` per utilizzare l'autenticazione JSON quando la sicurezza è abilitata.	

OAuth Configurazione

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-A0031	Warn	La proprietà non gapwalk-application.securit y.issuerUri deve essere impostata quando gapwalk-applicatio n.security.identity non lo è. oauth La configurazione attuale non è coerente. Impostate gapwalk-applicatio n.security.identity: oauth e configurate l'URI dell'emittente o rimuovete la proprietà gapwalk-application.security.issuerURI. La configurazione dell'URI dell'emittente è applicabile solo quando si utilizza l'autenticazione. OAuth	
BA-A0032	Warn	Proprietà gapwalk-applicatio n.security.issuerUri obbligatoria mancante. Questa proprietà è obbligatoria quando si utilizza OAuth l'autenticazione. Quando si utilizza OAuth authentication (gapwalk-application.securit y.identity: oauth), è necessario gapwalk-application.securit y.issuerUri impostare l'URL del proprio provider di identità (IdP).	
BA-A0033	Errore	Formato URI non valido per. gapwalk-application.securit y.issuerUri Il valore deve essere un URI valido. Imposta gapwalk-application.securit	

Chiave	Gravità	Testo	Dettagli aggiuntivi
		`y.issuerUri` su un URI valido del tuo provider di identità (IdP).	
`BA-A0041`	Warn	La proprietà non `gapwalk-application.securit y.allowedOrigins` deve essere impostata quando gapwalk-applicatio n.security.identity non lo è. `oauth` La configurazione attuale non è coerente. Impostate `gapwalk-applicatio n.security.identity: oauth` e configurate le origini consentite o rimuovete la proprietà Gapwalk-A pplication.Security.AllowedOrigins. OAuth La configurazione delle origini consentite è applicabile solo quando si utilizza l'autenticazione.	

Filtraggio URI

Chiave	Gravità	Testo	Dettagli aggiuntivi
`BA-A0101`	Fatale	Valore non valido per. `gapwalk-application.securit y.filterURIs` Solo `enabled` o `disabled` sono valori supportat i. Se non è impostato, il filtro URI sarà disabilitato per impostazione predefinita. `gapwalk-applicatio n.security.filterURIs: enabled`Impostare per attivare il filtraggio URI o `gapwalk-application.securit y.filterURIs: disabled`	

Chiave	Gravità	Testo	Dettagli aggiuntivi
		per disattivarlo. Rimuovi la proprietà per utilizzare il valore predefinito ()`disabled`.	
`BA-A0111`	Warn	La proprietà non `gapwalk-application.securit y.blockedURIs` deve essere impostata quando il filtro URI è disabilit ato. La configurazione corrente non è coerente. Abilita il filtraggio URI impostando o `gapwalk-applicatio n.filterURIs: enabled` rimuovendo la `gapwalk-applicatio n.security.blockedURIs` proprietà. La configurazione di blocco degli URI è applicabile solo quando il filtraggio URI è abilitato.	
`BA-A0112`	Warn	Proprietà `gapwalk-applicatio n.security.blockedURIs` obbligatoria mancante o vuota. Questa proprietà è obbligatoria e non deve essere vuota quando il filtro URI è abilitato. Impostato `gapwalk-application.securit y.blockedURIs` con un elenco separato da virgole da bloccare quando il URIs filtraggio URI è abilitato. Ad esempio: `gapwalk-applicatio n.security.blockedURIs: URI-1, URI-2, URI-3`	

OAuth Configurazione di sicurezza

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-A1010	Fatale	Si è verificato un errore durante la configurazione della catena OAuth di filtri di sicurezza. Controlla la tua OAuth configurazione, inclusa la registrazione del client, le impostazioni CORS e la protezione CSRF. Assicuratevi che tutte le proprietà richieste siano impostate correttamente nel file application-main.yml. Verifica che il tuo OAuth provider sia accessibile e configurato correttamente.	

AWS Errori di configurazione di Secrets Manager

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-A2000	Fatale	Impossibile analizzare il segreto delle leggi.	
BA-A2001	Fatale	Problema di autorizzazione AWS durante l'analisi segreta.	L'errore mostra l'errore sottostante. Spesso è correlato a un'autorizzazione mancante nel ruolo IAM utilizzato.
BA-A2002	Fatale	L'applicazione richiede l'accesso di rete al servizio AWS Secrets Manager per iniziare. Questo problema può essere risolto utilizzando un ambiente accessibile al pubblico o creando un gateway NAT PrivateLink o un gateway NAT.	Per informazioni, consulta Accesso Modernizzazione del mainframe AWS tramite un endpoint di AWS PrivateLink interfaccia.
BA-A2003	Fatale	Il nome DB per il segreto del database non è stato passato né nel segreto né	Per informazioni, consulta AWS I segreti di

Chiave	Gravità	Testo	Dettagli aggiuntivi
		nella chiave yaml. Aggiungi il nome del database alla configurazione, in un `dbname` campo del segreto o nella chiave yaml dedicata.	Transform for mainframe Runtime.

AWS Codici di errore di runtime Transform for mainframe relativi a Blusam

Blusamcodici di errore, con il prefisso. `BA-B`

Chiave	Gravità	Testo	Dettagli aggiuntivi
`BA-B0001`	Warn	Valore non valido per`bluesam.disabled` . `true`È supportata solo la disattivazioneBlusam. Il comportamento predefinito è che Bluesam sia abilitato. Imposta `bluesam.disabled: true` nella tua configurazione se vuoi disabilitarloBluesam. Rimuovi la proprietà per mantenerla Blusam abilitata.	
`BA-B0011`	Warn	BluesamLa cache non sarà abilitata. Imposta `bluesam.cache: ehcache` per usare EhCache o `bluesam.cache: redis` usare Redis. Qualsiasi altro valore disabiliterà Bluesam la cache.	
`BA-B0021`	Fatale	Blusamè attivo ma l'origine dati blusam non è definita nella configurazione. Imposta le chiavi di `datasource.blusamDs` configurazione (sottochiavi url, nome utente, password. ..) o usa un segreto. AWS	Per informazioni, consultare Configurazione di Blusam.

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-B2000	Warn	BlusamL'indice supera i limiti della memoria cache. Ignora la cache e persiste direttamente nel database.	
BA-B2001	Warn	Impossibile aggiungere l'Blusami ndice alla cache. Ignorando la cache e persistendo direttamente nel database.	
BA-B2002	Errore	Operazione non supportata: il calcolo della capacità della pagina non è supportato	MetadataPersistence l'implementazione non supporta il calcolo della capacità della pagina

AWS Codici di errore di runtime Transform for mainframe relativi ai batch

Codici di errore dei batch, con il prefisso BA-C

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-C1010	Warn	Il contesto dello script Job è nullo, il checkpoint procederà senza il contesto dello script.	
BA-C1011	Warn	Il contesto dello script Job non implementa Serializable, il checkpoint procederà senza il contesto dello script.	

AWS Codici di errore Transform for mainframe Runtime relativi a Datasimplifier

Codici di errore Datasimplifier, con il prefisso BA-D

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-D0001	Fatale	Riferimento di generazione GDG parametrizzato non valido. Usa il formato intero con segno o. (-nnn to +nnn) 0	
BA-D1001	Fatale	Unione vuota non valida: l'unione deve contenere almeno un campo figlio. Assicurati che il campo Union sia ben formato e segua la struttura prevista.	

AWS Transform for mainframe Runtime Codici di errore relativi ai file

Codici di errore dei file, con BA-F il prefisso. Questi errori sono correlati alle operazioni sui file, tra cui ESDS e GDG (Generation Data Groups).

GDG (Generation Data Groups)

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-F2000	Errore	Elaborazione dell'evento di eliminazi one GDG non riuscita. Verifica che il percorso del file GDG sia valido e che la coda degli eventi sia configurata correttamente.	
BA-F2001	Warn	Impossibile estrarre il nome del file dal percorso. Verifica che il formato del percorso del file GDG sia corretto.	

AWS Codici di errore di runtime Transform for mainframe relativi a CL Command Programs

CL Command pgmcodici di errore, con BA-H il prefisso.

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-H2010	Fatale	Il parametro FILE è richiesto nel comando RMVM. Fornire un parametro FILE valido al comando RMVM. Il parametro FILE specifica il file da cui rimuovere il membro.	
BA-H2011	Fatale	Tipo di parametro FILE imprevisto nel comando RMVM. Assicuratevi che il parametro FILE sia fornito come stringa o RecordAdaptable tipo. Controllate la sintassi del comando e i tipi di parametri passati a RMVM.	
BA-H2020	Warn	Il file non è un file di database. Operazione di blocco ignorata nel comando. È possibile bloccare solo i file di database. Il file è un file flat o non esiste. Verificate il nome del file e la libreria.	

AWS Codici di errore di runtime Transform for mainframe relativi a JICS

Codici di errore JICS, con il prefisso. BA-J

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-J0001	Fatale	Configurazione errata per JICS XA. DataSource Configura correttamente datasource.jicsDs.xa.data-source-class-name e altre proprietà xa oppure imposta su. spring.jta.enabled false	

AWS Codici di errore di runtime Transform for mainframe relativi a ADABAS

ADABAS-specificcodici di errore, con il prefisso. `BA-N`

Codici di errore dell'utilità ADASTRIP

Chiave	Gravità	Testo	Dettagli aggiuntivi
`BA-N1000`	Fatale	Immissione del `ADASTRIP` campo non valida: tipo previsto `'FieldFormat'` ma ricevuto. `'%s'`	Le serie di campi non sono supportate da `ADASTRIP` (Es: 'AA-AD').

AWS Codici di errore di runtime Transform for mainframe relativi alla coda

Codici di errore di coda, con il prefisso. `BA-Q`

Chiave	Gravità	Testo	Dettagli aggiuntivi
`BA-Q0001`	Warn	Valore non valido per`mq.connection.pool.share` . `true`È supportato solo per abilitare la condivisione del pool di connessioni JMS. Il comportamento predefinito è che la condivisione del pool di connessioni JMS è disabilitata. Imposta `mq.connection.pool.share: true` nella tua configurazione se desideri condividere il pool di connessioni JMS. Rimuovi la proprietà per mantenere disabilitata la condivisione del pool di connessioni JMS.	
`BA-Q1001`	Errore	Impossibile creare il pool di connessioni. Verificare la configurazione della connessione al pool e la disponibilità delle risorse e assicurarsi che vengano fornite le credenziali corrette.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-Q1002	Errore	Errore durante la chiusura della fabbrica. Verificate che la fabbrica non sia in uso e che tutte le risorse associate siano state rilasciate correttamente.	

AWS Codici di errore Redis Transform for mainframe Runtime

Codici di errore Redis, con il prefisso. BA-R

Errori di configurazione Redis

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0001	Fatale	Configurazione Redis mancante per la funzionalità tramite il percorso specificato. Specificate la configura zione nel percorso specificato o definite le impostazioni Redis globali.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0002	Fatale	Parametro di configurazione Redis mancante. Aggiungi il parametro richiesto.	
BA-R0003	Warn	Nessuna configurazione Redis esplicita trovata per il percorso. La configura zione Gapwalk Redis verrà utilizzata per la connessione. Puoi ignorarlo se lo desideri.	
BA-R0004	Fatale	Errore durante il recupero delle variabili di configurazione Redis dal percorso specificato. Controlla le impostazioni del file di configurazione e dell'ambiente.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0005	Fatale	Valore Redis mancante per il parametro specied. Controlla le impostazioni del file di configurazione e dell'ambiente.	

Errori di convalida della proprietà Redis

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0020	Errore	Errore di configurazione Redis: valore di configurazione Redis non valido per il parametro.	
BA-R0021	Fatale	Errore di configurazione Redis: modalità Redis non valida. Deve essere una delle due modalità specificate.	
BA-R0022	Errore	Errore nell'impostazione della proprietà. Controlla il formato e i vincoli del valore della proprietà.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0023	Errore	La chiave o il valore della proprietà è nullo per l'immissione nella mappa delle proprietà. Assicurati che tutte le chiavi e i valori delle proprietà non siano nulli.	
BA-R0024	Errore	Chiave di proprietà sconosciuta o non supportata.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0025	Errore	Valore numerico non valido per la proprietà. Assicuratevi che il valore possa essere convertito nel tipo numerico richiesto.	

Errori di connessione Redis

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0081	Fatale	Connessione al server Redis non riuscita: connessione rifiutata. Verifica che il server Redis sia in esecuzione e che host/port siano corretti.	
BA-R0082	Fatale	Timeout della connessione durante la connessione al server Redis. Verifica la connettività di rete o aumenta il timeout della connessione.	
BA-R0083	Fatale	Timeout di lettura durante la connessio ne al server Redis. Controlla le prestazioni del server Redis o aumenta il timeout di lettura.	
BA-R0084	Fatale	Errore SSL durante la connessione a Redis. Verifica le impostazioni e i certificati SSL.	
BA-R0085	Fatale	Impossibile stabilire la connessione Redis. Controlla lo stato e la configura zione del server Redis.	
BA-R0086	Fatale	Autenticazione non riuscita per la connessione Redis. Verifica le credenziali di autenticazione Redis.	
BA-R0087	Fatale	Eccezione dei dati Redis. Controlla i log del server Redis per maggiori dettagli.	
BA-R0088 - BA-R0089	Fatale	Errore imprevisto durante la connessio ne a Redis. Rivedi il messaggio di errore e la configurazione di Redis.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0090	Fatale	Impossibile creare la configurazione del pool Redis. Rivedi il messaggio di errore e la configurazione di Redis.	

Errori di configurazione specifici delle funzionalità

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0100	Errore	BlusamErrore di configurazione Blusam della cache: la cache è configurata per utilizzare redis (`bluesam.cache : redis`), mancano l'hostName e la configurazione della porta richiesti. Specificate la configurazione Redis nel percorso `bluesam.redis` o definite le impostazioni Redis globali utilizzan do. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `blusam.ca che` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0200	Errore	BlusamErrore di configurazione dei blocchi: Blusam locks è configurato per utilizzare redis (`bluesam.locks : redis`), mancano l'hostName e la configurazione della porta richiesti. Specificate la configurazione Redis nel percorso `bluesam.locks` o definite le impostazioni Redis globali utilizzan do. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `blusam.lo cks` proprietà come impostazione	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Chiave	Gravità	Testo	Dettagli aggiuntivi
		predefinita o impostala su un valore diverso.	
BA-R0300	Errore	Errore di configurazione di Gapwalk File Locks: Gapwalk file locks è configura to per utilizzare redis (`filesLock s.provider : redis`), mancano l'hostName richiesto e la configurazione della porta. Specificate la configura zione Redis nel percorso o definite le impostazioni Redis globali utilizzando. `filesLocks.redis` `gapwalk.r edis` Se non desideri utilizzare redis, rimuovi la `filesLocks.provider` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0400	Errore	Errore di configurazione del catalogo del set di dati: il catalogo del set di dati è configurato per utilizzar e redis (`datasimplifier.cat alogImplementation : redis`), mancano l'hostName e la configura zione della porta richiesti. Specificate la configurazione Redis nel percorso `datasimplifier.redis` o definite le impostazioni Redis globali utilizzando. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `datasimplifier.catalogImple mentation` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0500	Errore	Errore di configurazione della cache di sessione: la cache della sessione è configurata per utilizzare redis (spring.session.store-type : redis), mancano l'hostName e la configurazione della porta richiesti. Specificate la configurazione Redis nel percorso jics.redis o definite le impostazioni Redis globali utilizzando. gapwalk.redis Se non desideri utilizzare redis, rimuovi la spring.session.store-type proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0550	Errore	Errore di configurazione del Session Tracker: il tracker della sessione è configurato per utilizzare redis (session-tracker.store-type : redis), mancano l'hostName e la configurazione della porta richiesti . Specificate la configurazione Redis nel percorso session-tracker.redis o definite le impostazioni Redis globali utilizzando. gapwalk.redis Se non desideri utilizzare redis, rimuovi la session-tracker.store-type proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0600	Errore	Errore di configurazione delle definizioni delle risorse JICS: le definizioni delle risorse JICS sono configurate per utilizzare redis (`jics.resource-definitions.store-type : redis`), mancano l'hostName e la configurazione della porta richiesti. Specificate la configurazione Redis nel percorso `jics.redis` o definite le impostazioni Redis globali utilizzando. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `jics.resource-definitions.store-type` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0700	Errore	Errore di configurazione delle code JICS TS: le code JICS TS sono configurate per utilizzare redis (), mancano `jics.parameters.tsqimpl : redis` l'hostName richiesto e la configurazione della porta. Specificate la configurazione Redis nel percorso `jics.queues.ts.redis` o definite le impostazioni Redis globali utilizzando. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `jics.parameters.tsqimpl` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-R0800	Errore	Errore di configurazione del checkpoint JCL: il checkpoint JCL è configurato per utilizzare redis (`jcl.checkpoint.provider : redis`), mancano il nome host e la configurazione della porta richiesti. Specificate la configurazione Redis nel percorso `jcl.redis` o definite le impostazioni Redis globali utilizzando. `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `jcl.checkpoint.provider` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime
BA-R0900	Errore	Errore di configurazione Blu4iv Locks: Blu4iv locks è configurato per utilizzare redis (`blu4iv.lock : redis`), mancano l'hostName richiesto e la configurazione della porta. Specificate la configurazione Redis nel percorso o definite le impostazioni Redis globali utilizzando. `blu4iv.lock.redis` `gapwalk.redis` Se non desideri utilizzare redis, rimuovi la `blu4iv.lock` proprietà come impostazione predefinita o impostala su un valore diverso.	Proprietà della cache Redis disponibili in AWS Transform for mainframe Runtime

Errori Blusam Redis Cache

Errori di configurazione e connessione di Blusam Redis Cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1200	Errore	Impossibile inizializzare la cache Redis *******. Verifica i parametri di connessione e configurazione di Redis.	
BA-R1201	Errore	Impossibile chiudere la connessione Redis. Controlla lo stato della connessione Redis e verifica che sia stata precedentemente stabilita una connessione. Questo errore può verificarsi se si tenta di chiudere una connessione inesistente.	

Operazioni sulla cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1300	Errore	Impossibile recuperare la cache per il nome. Verifica il nome della cache e la connessione Redis.	
BA-R1301	Errore	Impossibile recuperare gli indici della cache per il nome. Verifica il nome della cache e la connessione Redis.	

Gestione dei metadati

Chiave	Gravità	Testo	Dettagli aggi
BA-R1400	Errore	Impossibile recuperare i metadati per la cache. Controlla la chiave dei metadati e la connessione Redis.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R1401	Errore	Impossibile memorizzare i metadati per la cache. Verifica l'oggetto dei metadati e la connessione Redis.	
BA-R1402	Errore	Impossibile rimuovere i metadati dalla cache. Controlla il nome della cache e la connessione Redis.	

Riscaldamento della cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1500	Errore	Impossibile controllare lo stato di riscaldamento della cache. Verifica il nome della cache e la connessione Redis.	
BA-R1501	Errore	Impossibile contrassegnare la cache come riscaldat a. Controlla il nome della cache e la connessione Redis.	

Operazioni Publisher/Subscriber

Chiave	Gravità	Testo	Dettagli aggi
BA-R1600	Errore	Impossibile creare il listener di aggiornamento dei metadati per la cache. Controlla le pub/sub proprietà Redis: subscriptionsPerConnection (impostazione predefinita: 10), subscriptionConnectionPool Size (impostazione predefinita: 100), ConnectTimeout (impostazione predefinita: 10000).	
BA-R1601	Errore	Impossibile pubblicare l'aggiornamento dei metadati per la cache. Controlla le pub/sub proprietà di Redis:	

Chiave	Gravità	Testo	Dettagli aggi
		ConnectTimeout (impostazione predefinita: 10000), ReadTimeout (impostazione predefinita: 2000).	
BA-R1602	Errore	Impossibile creare il listener di invalidazione della cache per la cache. Controlla le pub/sub proprietà di Redis e regolale in base al carico di invalidazione.	
BA-R1603	Errore	Impossibile pubblicare l'invalidazione della cache per la cache. Valuta la possibilità di aumentare i valori di timeout per set di invalidazione di grandi dimensioni.	

Operazioni di registrazione della cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1710	Errore	Impossibile recuperare il record dalla cache. Verifica che il record esista e che la connessione Redis sia attiva.	
BA-R1711	Errore	Impossibile memorizzare il record nella cache. Controlla la connessione Redis e registra i dati.	
BA-R1712	Errore	Impossibile memorizzare più record nella cache. Verifica le impostazioni della connessione Redis e del funzionamento in batch.	
BA-R1720	Errore	Operazione Batch non riuscita durante la scrittura impaginata. Controlla le proprietà pageSizeIn Bytes e ConnectTimeout.	
BA-R1721	Errore	Impossibile eseguire l'operazione in blocco. Verifica lo stato del cluster Redis e le impostazioni di connessione.	
BA-R1730	Errore	Impossibile eliminare il record dalla cache. Verifica l'esistenza del record e la connessione Redis.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R1731	Errore	Operazione di eliminazione in batch parzialmente fallita. Verifica lo stato del cluster Redis e l'esistenza del record.	
BA-R1732	Errore	Impossibile cancellare la cache. Controlla la connessione Redis e il nome della cache.	

Operazioni sull'indice della cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1801	Errore	Impossibile recuperare l'indice dalla cache. Verifica che l'indice esista e che la connessione Redis sia attiva.	
BA-R1802	Errore	Impossibile memorizzare l'indice nella cache. Verifica l'integrità della connessione Redis e dei dati dell'indi ce.	
BA-R1803	Errore	È stato rilevato un formato di chiave non valido durante l'elaborazione delle chiavi di indice nella cache. Verifica la presenza di dati danneggiati o schemi chiave non validi.	
BA-R1804	Errore	Impossibile recuperare le chiavi dell'indice dalla cache. Verifica la connessione Redis e il nome della cache.	
BA-R1805	Errore	Impossibile eliminare l'indice dalla cache. Verifica che l'indice esista e la connessione Redis.	
BA-R1811	Warn	Elaborazione saltata di una chiave di indice non valida. Verifica il formato della chiave e l'integrità dei dati.	

Operazioni di invalidazione della cache

Chiave	Gravità	Testo	Dettagli aggi
BA-R1900	Errore	Impossibile inizializzare il listener di invalidazione della cache per il set di dati. Verifica i parametri di sottoscrizione nella configurazione.	
BA-R1901	Errore	Errore nell'elaborazione del messaggio di invalidaz ione della cache per il set di dati. Controlla l'impleme ntazione da parte dei consumatori e il formato dei messaggi.	
BA-R1902	Errore	È stato ricevuto un messaggio di invalidazione non valido per il set di dati. Verifica il formato e il contenuto del messaggio.	
BA-R1903	Errore	Impossibile mantenere la connessione dell'abbo nato per il set di dati. Controlla i parametri dell'abbo namento Redis e le impostazioni di connessione.	

Codici di errore Blusam Redis Locks

Operazioni di blocco

Chiave	Gravità	Testo	Dettagli aggi
BA-R2300	Errore	Parametri non validi per il funzionamento del blocco Bluesam. Verifica che il nome del set di dati e i parametri di blocco non siano nulli.	
BA-R2301	Errore	Impossibile salvare il blocco del set di dati. Controlla la connessione Redis e lo stato di blocco.	
BA-R2302	Errore	Impossibile salvare il blocco dei record per il set di dati. Verifica che il record esista e che la connessio ne Redis sia attiva.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R2303	Errore	Impossibile rimuovere il blocco del set di dati. Controlla se il blocco esiste e la connessione Redis è attiva.	
BA-R2304	Errore	Impossibile rimuovere il blocco dei record per il set di dati. Verifica che il record e il blocco esistano.	

Blocca le operazioni sullo stato

Chiave	Gravità	Testo	Dettagli aggi
BA-R2310	Errore	Parametri di pulizia dello stato di blocco non validi. Fornire un valore positivo per il parametro MaxAge.	
BA-R2311	Warn	Avviso di integrità dello stato di blocco. Verifica la presenza di danni ai dati negli stati di blocco.	
BA-R2312	Errore	Impossibile pulire lo stato di blocco. Verifica la connessione Redis e l'integrità dello stato.	
BA-R2313	Errore	Operazione di pulizia dello stato di blocco non riuscita. Controlla la connessione Redis e la memoria disponibile.	
BA-R2314	Errore	Impossibile ottenere lo stato del set di dati. Controlla la connessione Redis e l'esistenza del set di dati.	
BA-R2315	Errore	Impossibile ottenere lo stato del record per il set di dati. Verifica che il record esista e che la connessio ne Redis sia attiva.	
BA-R2316	Errore	Impossibile controllare l'autorizzazione di scrittura del set di dati. Controlla la connessione Redis e lo stato di blocco.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R2317	Errore	Impossibile controllare l'autorizzazione di scrittura dei record per il set di dati. Verifica lo stato del record e del blocco.	
BA-R2318	Errore	Impossibile controllare l'autorizzazione di lettura del set di dati. Controlla la connessione Redis e lo stato di blocco.	
BA-R2319	Errore	Impossibile controllare l'autorizzazione di lettura dei record per il set di dati. Verifica lo stato del record e del blocco.	
BA-R2320	Errore	Impossibile salvare lo stato. Verifica la connessione Redis e l'integrità dei dati dello stato.	

Errori di timeout di blocco

Chiave	Gravità	Testo	Dettagli aggi
BA-R2330	Errore	Timeout in attesa del permesso di scrittura del set di dati. Valuta la possibilità di aumentare il timeout di blocco o di verificare la presenza di deadlock.	
BA-R2331	Errore	Timeout durante l'acquisizione del blocco per il set di dati. Verifica se ci sono operazioni di lunga durata che tengono premuto il blocco.	
BA-R2332	Errore	Interrotto in attesa del blocco del set di dati. Verifica la presenza di interruzioni dei thread o problemi di sistema.	
BA-R2333	Errore	Timeout durante l'acquisizione del blocco per la registrazione nel set di dati. Prendi in considerazione l'aumento del timeout di blocco o ottimizza i modelli di accesso ai record.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R2334	Errore	Interrotto in attesa del blocco del record nel set di dati. Analizza le interruzioni dei thread o i problemi a livello di sistema.	

GWFileBlocca i codici di errore

Errori di configurazione e inizializzazione

Chiave	Gravità	Testo	Dettagli aggi
BA-R3099	Errore	Impossibile chiudere il client GWFile Locks Redisson. Verifica le operazioni attive e riprova a spegnere.	
BA-R3101	Fatale	Impossibile creare il client di redissione per GapWalk i blocchi dei file. Verifica la tua configurazione su 'FilesLocks.redis' o le impostazioni Redis globali 'gapwalk.redis', se utilizzate.	
BA-R3102	Fatale	Il client Redisson non può essere nullo per Files Locks. GapWalk Verifica la tua configurazione su 'FilesLocks.redis' o le impostazioni Redis globali 'gapwalk.redis' se utilizzate.	

GWFileBlocca le operazioni

Chiave	Gravità	Testo	Dettagli aggi
BA-R3200	Warn	I metadati di blocco sono nulli per il file. Ciò può causare incoerenze tra i dati e portare a errori di blocco o situazioni di stallo. Verifica l'integrità dei dati Redis e verifica la presenza di metadati di blocco. outdated/corrupted	

Chiave	Gravità	Testo	Dettagli aggi
BA-R3210	Errore	Impossibile controllare lo stato del blocco del file. Controlla la connessione Redis e il percorso del file.	
BA-R3220	Errore	L'acquisizione del blocco è stata interrotta per il file. Verifica la presenza di interruzioni nei thread.	
BA-R3230	Errore	Impossibile memorizzare i metadati di blocco per il file. Verifica la connessione Redis e la disponibilità della memoria.	
BA-R3240	Informazioni	Il file è bloccato dal processo. Attendi il completamento dell'operazione corrente.	
BA-R3250	Errore	Impossibile bloccare il file. Controlla la connessione Redis e la configurazione del blocco.	
BA-R3260	Errore	Impossibile rilasciare il blocco per il file. Verifica che il blocco esista e che la connessione sia attiva.	
BA-R3270	Errore	Impossibile rilasciare i blocchi per il lavoro. Controlla il contesto del lavoro e la connessione Redis.	
BA-R3280	Errore	Impossibile recuperare i blocchi per il contesto lavorativo. Verifica l'esistenza del contesto di lavoro e della connessione Redis.	
BA-R3290	Errore	Blocca il danneggiamento dei metadati rilevato per il file. Verifica l'integrità dei dati Redis.	

Codici di errore del catalogo Dataset

Operazioni sul set di dati

Chiave	Gravità	Testo	Dettagli aggi
BA-R4200	Errore	Registrazione del set di dati non riuscita. Controlla la connessione Redis e i dati del set di dati.	
BA-R4201	Errore	Impossibile verificare l'esistenza del set di dati. Verifica l'identificatore del set di dati e la connessione Redis.	
BA-R4220	Errore	Impossibile recuperare il set di dati. Controlla l'identif icatore del set di dati e la connessione Redis.	
BA-R4221	Errore	Impossibile recuperare tutti i set di dati. Verifica la connessione Redis e riprova l'operazione.	
BA-R4300	Errore	Impossibile rimuovere il set di dati. Controlla la connessione Redis e l'esistenza del set di dati.	

Codici di errore Session & Session Tracker

Operazioni di sessione

Chiave	Gravità	Testo	Dettagli aggi
BA-R5050	Errore	Impossibile ottenere la connessione Redis per il tracker della sessione. Controlla le impostazioni della connessione Redis e lo stato del server.	
BA-R5051	Errore	Impossibile deserializzare i dati della sessione. Verifica il formato e l'integrità dei dati della sessione.	
BA-R5052	Errore	Impossibile recuperare le sessioni per il nodo. Controlla l'ID del nodo e la connessione Redis.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R5053	Errore	Impossibile mantenere la sessione. Verifica i dati della connessione e della sessione Redis.	
BA-R5054	Errore	Impossibile pulire la sessione. Controlla l'ID della sessione e la connessione Redis.	
BA-R5055	Warn	Nessuna sessione trovata per il nodo. Verifica l'ID del nodo o le sessioni potrebbero essere scadute.	
BA-R5056	Errore	Impossibile serializzare l'elenco delle sessioni in JSON. Verifica l'integrità dei dati della sessione.	
BA-R5057	Errore	Parametro di funzionamento del tracker di sessione non valido. Verificate che siano stati forniti tutti i parametri obbligatori.	
BA-R5058	Errore	Operazione batch non riuscita per le sessioni. Controlla la connessione Redis e riprova l'operazi one.	
BA-R5059	Warn	Avviso di scadenza della sessione. Potrebbe essere necessario modificare il timeout della sessione.	

Definizioni delle risorse JICS, codici di errore

Codici di errore Redis, con il prefisso BA-R.

Errori di configurazione

Chiave	Gravità	Testo	Dettagli aggi
BA-R6101	Errore	Impossibile aggiornare la risorsa Redis con ID. Controlla la connessione Redis e verifica le autorizza zioni di scrittura.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R6102	Errore	Impossibile recuperare la risorsa da Redis. Controlla la connessione Redis e verifica che la risorsa esista.	
BA-R6110	Errore	Impossibile eseguire l'operazione sul modello Redis. Controlla la connessione Redis e lo stato del server.	

Operazioni con l'elenco ICS

Chiave	Gravità	Testo	Dettagli aggi
BA-R6200	Errore	Elenco JICS non trovato nella cache Redis. Verifica il nome dell'elenco e assicurati che esista nella cache Redis.	
BA-R6201	Errore	Impossibile recuperare l'elenco JICS da Redis. Controlla la connessione Redis e verifica che l'elenco esista.	
BA-R6202	Errore	Impossibile recuperare tutti gli elenchi JICS. Controlla la connessione Redis e lo stato del server.	
BA-R6203	Errore	Impossibile mantenere l'elenco JICS. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6204	Errore	Impossibile eliminare l'elenco JICS. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6205	Errore	Impossibile trovare l'elenco per ID. Verifica l'ID dell'elenco e assicurati che la connessione Redis funzioni.	
BA-R6206	Errore	Impossibile recuperare l'elenco con i bambini. Controlla la connessione Redis e verifica che l'elenco esista.	

Operazioni del gruppo JICS

Chiave	Gravità	Testo	Dettagli aggi
BA-R6300	Errore	Gruppo JICS non trovato nella cache Redis. Verifica il nome del gruppo e assicurati che esista nella cache Redis.	
BA-R6301	Errore	Impossibile recuperare il gruppo JICS con elenchi. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6302	Errore	Elenco referenziato dal gruppo non trovato. Verifica che l'elenco esista prima di associarlo a un gruppo.	
BA-R6303	Errore	Impossibile recuperare l'elenco dei gruppi per l'elenco JICS. Controlla la connessione Redis e verifica che l'elenco esista.	
BA-R6304	Errore	Impossibile recuperare per il gruppo. Controlla la connessione Redis e verifica che le risorse esistano.	
BA-R6305	Errore	Impossibile recuperare il gruppo con tutti i bambini. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6306	Errore	Impossibile recuperare tutti i gruppi JICS. Controlla la connessione Redis e lo stato del server.	
BA-R6307	Errore	Impossibile trovare i gruppi nell'elenco. Controlla la connessione Redis e verifica l'esistenza dell'elenco.	
BA-R6308	Errore	Impossibile verificare se il gruppo è presente. Controlla la connessione Redis e lo stato del server.	
BA-R6309	Errore	Impossibile contare i gruppi. Controlla la connessione Redis e lo stato del server.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R6310	Errore	Impossibile aggiungere gruppi all'elenco. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6311	Errore	Impossibile mantenere il gruppo JICS. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6312	Errore	Eliminazione del gruppo non riuscita. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6313	Errore	Impossibile eliminare il gruppo JICS. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	

Operazioni sui file JICS

Chiave	Gravità	Testo	Dettagli aggi
BA-R6400	Errore	File non trovato nel gruppo. Verifica il nome del file e assicurati che esista nel gruppo specificato.	
BA-R6401	Errore	Impossibile ottenere l'ID del file per il gruppo. Controlla la connessione Redis e verifica l'esistenza del file.	
BA-R6402	Errore	Impossibile recuperare il file IDs attivo. Controlla la connessione Redis e verifica l'esistenza del file.	
BA-R6403	Errore	Impossibile trovare i file per nome. Controlla la connessione Redis e verifica l'esistenza dei file.	
BA-R6404	Errore	Impossibile trovare il file per ID. Controlla la connessione Redis e verifica l'esistenza del file.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R6405	Errore	Impossibile trovare i file nel gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6406	Errore	Impossibile recuperare il primo file dal gruppo. Controlla la connessione Redis e verifica l'esistenza del file.	
BA-R6407	Errore	Impossibile rendere persistente il file. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6408	Errore	Impossibile eliminare il file con ID. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6409	Errore	File con ID non trovato per l'eliminazione. Verifica che l'ID del file esista prima di tentare di eliminarlo.	
BA-R6410	Errore	Impossibile inizializzare l'elenco dei file per il gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	

Operazioni del programma JICS

Chiave	Gravità	Testo	Dettagli aggi
BA-R6500	Errore	Programma non trovato nel gruppo. Verifica il nome del programma e assicurati che esista nel gruppo specificato.	
BA-R6501	Errore	Impossibile recuperare l'ID del programma per il gruppo. Controlla la connessione Redis e verifica l'esistenza del programma.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R6502	Errore	Impossibile ottenere il programma IDs attivo. Controlla la connessione Redis e verifica l'esistenza del programma.	
BA-R6503	Errore	Impossibile trovare i programmi per nome. Controlla la connessione Redis e verifica l'esistenza dei programmi.	
BA-R6504	Errore	Impossibile trovare il programma per ID. Controlla la connessione Redis e verifica l'esistenza del programma.	
BA-R6505	Errore	Impossibile trovare i programmi nel gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6506	Errore	Impossibile recuperare il primo programma dal gruppo. Controlla la connessione Redis e verifica l'esistenza del programma.	
BA-R6507	Errore	Impossibile rendere persistente il programma. Controlla la connessione Redis e verifica le autorizza zioni di scrittura.	
BA-R6508	Errore	Impossibile eliminare il programma con ID. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6509	Errore	Programma con ID non trovato per l'eliminazione. Verifica che l'ID del programma esista prima di tentare l'eliminazione.	
BA-R6510	Errore	Impossibile inizializzare l'elenco dei programmi per il gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	

Operazioni di transazione JICS

Chiave	Gravità	Testo	Dettagli aggi
BA-R6600	Errore	Transazione non trovata nel gruppo. Verifica il nome della transazione e assicurati che esista nel gruppo specificato.	
BA-R6601	Errore	Impossibile ottenere l'ID della transazione per il gruppo. Controlla la connessione Redis e verifica l'esistenza della transazione.	
BA-R6602	Errore	Impossibile ottenere una transazione IDs attiva. Controlla la connessione Redis e verifica l'esistenza della transazione.	
BA-R6603	Errore	Impossibile trovare le transazioni per nome. Controlla la connessione Redis e verifica che le transazioni esistano.	
BA-R6604	Errore	Impossibile trovare la transazione per ID. Controlla la connessione Redis e verifica l'esistenza della transazione.	
BA-R6605	Errore	Impossibile trovare le transazioni nel gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6606	Errore	Transazione persistente non riuscita. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6607	Errore	Impossibile eliminare la transazione con ID. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6608	Errore	Impossibile inizializzare l'elenco delle transazioni per il gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	

Operazioni JICS TDQueue

Chiave	Gravità	Testo	Dettagli aggi
BA-R6700	Errore	TDQueue non trovato nel gruppo. Verifica il TDQueue nome e assicurati che esista nel gruppo specificato.	
BA-R6701	Errore	Impossibile ottenere l' TDQueue ID per il gruppo. Controlla la connessione Redis e verifica che TDQueue esista.	
BA-R6702	Errore	Impossibile TDQueue IDs attivarsi. Controlla la connessione Redis e verifica che TDQueue esista.	
BA-R6703	Errore	Impossibile trovare TDQueues per nome. Controlla la connessione Redis e verifica che TDQueues esista.	
BA-R6704	Errore	Impossibile trovare TDQueue per ID. Controlla la connessione Redis e verifica che TDQueue esista.	
BA-R6705	Errore	Impossibile trovare TDQueues nel gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6706	Errore	Impossibile TDQueue persistere. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6707	Errore	Eliminazione TDQueue con ID non riuscita. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6708	Errore	Impossibile inizializzare l' TDQueue elenco per il gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	

Operazioni JICS TSModel

Chiave	Gravità	Testo	Dettagli aggi
BA-R6800	Errore	TSModel non trovato nel gruppo. Verifica il TSModel nome e assicurati che esista nel gruppo specificato.	
BA-R6801	Errore	Impossibile ottenere l' TSModel ID per il gruppo. Controlla la connessione Redis e verifica che TSModel esista.	
BA-R6802	Errore	Impossibile TSModel IDs attivarsi. Controlla la connessione Redis e verifica che TSModel esista.	
BA-R6803	Errore	Impossibile trovare TSModels per nome. Controlla la connessione Redis e verifica che TSModels esista.	
BA-R6804	Errore	Impossibile trovare TSModel per ID. Controlla la connessione Redis e verifica che TSModel esista.	
BA-R6805	Errore	Impossibile trovare TSModels nel gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	
BA-R6806	Errore	Impossibile TSModel persistere. Controlla la connessione Redis e verifica le autorizzazioni di scrittura.	
BA-R6807	Errore	Eliminazione TSModel con ID non riuscita. Controlla la connessione Redis e verifica le autorizzazioni di eliminazione.	
BA-R6808	Errore	Impossibile inizializzare l' TSModel elenco per il gruppo. Controlla la connessione Redis e verifica l'esistenza del gruppo.	

Errori di autenticazione

Chiave	Gravità	Testo	Dettagli aggi
BA-R6900	Errore	Impossibile ottenere il nome utente connesso. Controlla il contesto di sicurezza e lo stato di autenticazione.	
BA-R6901	Errore	Errore di autenticazione durante l'accesso alla risorsa Redis. Verifica le autorizzazioni utente e la configura zione dell'autenticazione.	

Codici di errore TS Queue Redis

Operazioni generali di TS Queue

Chiave	Gravità	Testo	Dettagli aggi
BA-R7000	Warn	Avviso di funzionamento della coda TS: recupero dal registro con ambito di attività anziché dallo storage Redis. Configura lo storage Redis TS Queue se desideri utilizzarlo.	

Errori di configurazione

Chiave	Gravità	Testo	Dettagli aggi
BA-R7110	Errore	Errore di inizializzazione Jics TS Queue Redis: impossibile inizializzare le operazioni di elenco. Verifica la configurazione del modello Redis.	

Operazioni sulle code

Chiave	Gravità	Testo	Dettagli aggi
BA-R7711	Errore	Impossibile leggere dalla coda TS. Verifica dell'esis tenza della coda e dell'integrità dei dati.	
BA-R7712	Errore	Impossibile scrivere su TS Queue. Controlla la connessione Redis e le autorizzazioni di scrittura.	
BA-R7713	Errore	Impossibile eliminare TS Queue. Verifica l'esistenza della coda ed elimina le autorizzazioni.	
BA-R7714	Errore	Operazione dell'elenco delle code TS non riuscita. Verifica la disponibilità delle operazioni dell'elenco Redis.	
BA-R7715	Errore	Accesso all'indice non valido per TS Queue. Verifica che l'indice rientri nei limiti.	
BA-R7716	Warn	Avviso di funzionamento della dimensione della coda TS. Verifica l'esistenza e la connessione della coda.	

Codici di errore JCL Checkpoint di AWS Transform for mainframe Runtime

Errori di configurazione

Chiave	Gravità	Testo	Dettagli aggi
BA-R8110	Errore	Errore di inizializzazione JCL Checkpoint Redis: impossibile inizializzare le operazioni di elenco. Verifica la configurazione del modello Redis.	

Operazioni di recupero dei checkpoint

Chiave	Gravità	Testo	Dettagli aggi
BA-R8200	Errore	Impossibile recuperare i checkpoint JCL per il lavoro. Controlla la connessione Redis e il nome del lavoro.	
BA-R8210	Errore	Errore nel recupero dei checkpoint JCL dalla persistenza. Controlla la connessione Redis e il nome del lavoro.	
BA-R8600	Warn	Nessun checkpoint trovato per il lavoro. Verifica il nome del lavoro e l'esistenza del checkpoint.	

Operazioni di persistenza dei checkpoint

Chiave	Gravità	Testo	Dettagli aggi
BA-R8300	Errore	Impossibile mantenere i checkpoint JCL per il lavoro. Verifica la connessione Redis e le autorizzazioni di scrittura.	
BA-R8305	Errore	Impossibile impostare la scadenza per i checkpoin t delle offerte di lavoro. Verifica le impostazioni di connessione e scadenza di Redis.	
BA-R8400	Errore	La deserializzazione del checkpoint non è riuscita. Verifica l'integrità dei dati del checkpoint.	

Operazioni di Checkpoint Purge

Chiave	Gravità	Testo	Dettagli aggi
BA-R8500	Errore	Impossibile eliminare i checkpoint JCL dalla persisten za redis. Controlla le autorizzazioni di connessione e cancellazione di Redis.	

Chiave	Gravità	Testo	Dettagli aggi
BA-R8501	Errore	Impossibile eliminare il checkpoint JCL dalla persistenza redis. Controlla le autorizzazioni di connessione e cancellazione di Redis.	

Codici di errore Blu4iv Lock

Operazioni di blocco

Chiave	Gravità	Testo	Dettagli aggi
BA-R9200	Errore	Impossibile salvare il blocco Blu4iv per la registraz ione. Controlla la connessione Redis e blocca i dati.	
BA-R9201	Errore	Impossibile controllare lo stato del lock repository. Verifica lo stato della connessione e del repository Redis.	
BA-R9210	Errore	Impossibile verificare l'autorizzazione al blocco. Verifica le impostazioni di connessione e timeout di Redis.	
BA-R9220	Errore	Impossibile rimuovere il blocco Blu4iv. Controlla la connessione Redis e l'esistenza del blocco.	
BA-R9225	Errore	Impossibile cancellare i blocchi Blu4iv. Controlla la connessione e le autorizzazioni Redis.	
BA-R9900	Warn	Il lucchetto esiste già a fini di registrazione. Verifica lo stato del blocco e l'identificatore del record.	

AWS Codici di errore di runtime Transform for mainframe relativi ai programmi di utilità

Codici di errore di Utility pgm, con il prefisso. BA-U

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U1001	Warn	La scala dei secondi lilian in ingresso supera la scala massima. Al suo posto verrà utilizzata la scala massima.	
BA-U2001	Errore	Rilevata TrailerItem sottoclasse di elaborazione non riuscita. Assicurati che l'articolo del trailer sia supportato.	
BA-U2002	Errore	Analisi dell' TrailerItem argomento non riuscita. Assicurati che gli argomenti dell'articolo del trailer siano supportati.	
BA-U2003	Errore	Impossibile elaborare l'argomento del formato della data DFSORT. Assicurati che l'argomento relativo al formato della data sia supportato.	
BA-U2004	Errore	Input non valido durante l'estrazione dell'Header1 da. SortGrammar Assicurat i che la stringa Header1 sia ben formata e segua la struttura prevista.	
BA-U2005	Errore	Errore imprevisto durante l'elaborazione. SortGrammar Esamina i dettagli delle eccezioni e verifica l'integrità dell'input e la logica di analisi grammaticale.	
BA-U2006	Errore	Codice di feedback CEE2 F3: Impossibi le recuperare la data e l'ora correnti del sistema. Controlla la configurazione dell'orologio di sistema e le impostazioni del fuso orario.	
BA-U2007	Errore	Codice di feedback CEE2 EC: Impossibi le analizzare il valore della data passato	

Chiave	Gravità	Testo	Dettagli aggiuntivi
		a CEEDAYS o CEESECS. Assicurati che il formato della data sia supportato.	
BA-U2008	Errore	Codice di feedback CEE2 EH: l'input date/time in una chiamata CEEISEC, CEEDAYS o CEESECS non rientrava nell'intervallo valido. Assicurati che la data di input rientri nell'intervallo supportato.	
BA-U2009	Errore	Codice di feedback CEE2 EM: analisi di una stringa di immagine di input durante una chiamata a un date/time servizio. Assicurati che il formato dell'immagine sia supportato.	
BA-U2010	Errore	Codice di feedback CEE2 EG: Il valore della data di input di Lilian non rientrava nell'intervallo valido. Assicurati che la data lilian di input rientri nell'intervallo supportato.	
BA-U2011	Fatale	Il numero di argomenti ricevuti non corrisponde al numero di argomenti previsto. Assicurati che venga passato il numero corretto di argomenti al programma chiamato.	
BA-U2012	Errore	Elaborazione dell'argomento DFSORT TIME non riuscita. Verificare che il formato dell'argomento sia corretto.	
BA-U2013	Errore	Elaborazione dell'argomento DFSORT TIMENS non riuscita. Verificare che il formato dell'argomento sia corretto.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U2014	Errore	Caso OVERLAY non implementato. Consulta le specifiche OVERLAY o contatta l'assistenza per ricevere assistenza sull'implementazione.	
BA-U2015	Errore	Conversione della data giuliana non riuscita. Assicurati che il formato della data giuliano sia valido e contenga valori numerici.	
BA-U3001	Errore	Si è verificato un errore durante l'analisi di SYSIN per i programmi INZUTILB. Verifica la sintassi SYSIN e contatta l'assistenza se deve essere supportata.	
BA-U3002	Errore	Impossibile rilevare il tipo di database. L'impostazione predefinita è PostgreSQL. Verifica la connessione al database, la configurazione del driver e assicurati che sia caricato il driver JDBC corretto.	
BA-U3003	Errore	Operazione di scaricamento INZUTILB non riuscita. Rivedi la sintassi dei comandi SYSIN per l'utilità INZUTILB, verifica la connettività del database.	
BA-U3004	Errore	L'input SYSIN è vuoto o non contiene comandi validi. Aggiungi almeno un comando valido (UNLOAD, SELECT, ecc.) a SYSIN.	
BA-U3005	Errore	Il parametro richiesto OUTDDN è mancante. Specificare il set di dati di output utilizzando il comando OUTDDN (ddname) in SYSIN.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U3006	Errore	Il tipo di database non è supportato. Usa uno dei database supportati: PostgreSQL, Oracle o. DB2	
BA-U3007	Errore	Nessun formattatore trovato per il tipo di database durante l'esecuzione del programma INZUTILB. Analizza se il tipo deve essere considerato e contatta l'assistenza per l'analisi.	
BA-U3008	Errore	Formattazione non gestita per il formato di output selezionato, utilizzato invece il formato DSNTIAUL. Controlla il formato di output nella scheda di controllo o contatta l'assistenza se è necessaria un'implementazione.	
BA-U3009	Errore	Parametro di maschera non valido nell'opzione PIC. Verifica la tua scheda di controllo e usa una maschera valida.	
BA-U3010	Errore	Parametro di segno non valido nell'opzione PIC. I valori accettati sono '-', '+', 'P' e 'V'. Verifica la tua scheda di controllo e utilizza un parametro valido.	
BA-U3012	Errore	Impossibile analizzare il programma SYSIN for INZUTILB. Controlla le eccezioni di cui sopra e contatta l'assistenza per l'analisi.	
BA-U3013	Errore	Operazione INFUTILB Unload non riuscita. Rivedi la sintassi dei comandi SYSIN per l'utilità INFUTILB, verifica la connettività del database.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U3014	Errore	Tipo di file di output non valido per SYSPUNCH durante l'esecuzione del programma INFUTILB. Prendi in considerazione l'utilizzo di uno dei seguenti tipi: Configuration, o. GDGSupport FileSystemFileConf iguration DummyFileConfiguration	
BA-U3015	Errore	Nessun formattatore trovato per il tipo di database durante l'esecuzione del programma INFUTILB. Analizza se il tipo deve essere considerato e contatta l'assistenza per l'analisi.	
BA-U3016	Errore	Dimensione del record mancante per SYSPUNCH durante l'esecuzione del programma INFUTILB. Imposta la dimensione del record nella configura zione del file o aggiungilo al catalogo.	
BA-U3017	Errore	Tipo di file di output non valido per SYSREC. Prendi in considerazione l'utilizzo di uno dei seguenti tipi: Configuration o. GDGSupport FileSyste mFileConfiguration	
BA-U3018	Errore	Query SQL non supportata. Controlla la query SQL.	
BA-U3019	Errore	Nome di colonna non supportato. Controlla la definizione della tabella.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U3020	Errore	Opzione DFSIGDCB: impossibile aggiornare LRECL del file SYSREC. Prendi in considerazione l'utilizzo dell'opzione type o rimuovi DFSIGDCB. FileSystemFileConfiguration	
BA-U3021	Errore	La suddivisione dello scaricamento in blocchi non è supportata dal tipo di database corrente. Utilizza uno dei database supportati: PostgreSQL o Oracle.	
BA-U4001	Errore	Errore di analisi IDCAMS. Il contenuto SYSIN contiene una sintassi non valida o comandi non supportati. Esamina il contenuto SYSIN per eventuali errori di sintassi o errori di battitura. Controlla il messaggio di errore per i dettagli su ciò che non è stato analizzato.	
BA-U5001	Errore	Opzione POSITION non valida quando si chiama l'utilità DSNUTILB LOAD. Utilizzate un'opzione POSITION valida.	
BA-U5002	Errore	Opzione NULLIF non valida quando si chiama l'utilità DSNUTILB LOAD. Verifica che l'opzione NULLIF specifica ta sia valida e contatta l'assistenza se deve essere supportata.	
BA-U5003	Errore	Definizione di colonna non valida quando si chiama l'utilità DSNUTILB LOAD. Controlla la sintassi e contatta l'assistenza se deve essere supportata.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U5004	Errore	Tipo di dati di colonna non supportat o quando si chiama l'utilità DSNUTILB LOAD. Verifica che il tipo di dati specificato sia valido e contatta l'assiste nza se deve essere supportato.	
BA-U5005	Errore	Set di dati non definito o non valido quando si chiama l'utilità DSNUTILB COPY. Verificate che il set di dati sia specificato correttamente.	
BA-U5006	Errore	Si è verificato un errore durante l'elimina zione di un file durante la chiamata dell'utilità DSNUTILB COPY. Controlla i diritti di accesso del file.	
BA-U7010	Errore	DANTIAD: SYSPRINT DD non configura to. Verificare l'istruzione SYSPRINT DD.	
BA-U7020	Errore	DSNTIAD: esecuzione delle istruzioni SQL non riuscita. Rivedi la sintassi SQL.	
BA-U7030	Errore	DSNTIAD: I/O errore durante l'esecuzi one. Controlla il file SYSPRINT.	
BA-U7040	Errore	DSNTIAD: inizializzazione del writer SYSPRINT non riuscita. Controlla le autorizzazioni del file system.	
BA-U7050	Errore	DSNTIAD: impossibile scrivere su SYSPRINT. Controlla lo spazio su disco e le autorizzazioni dei file.	
BA-U7060	Errore	DSNTIAD: Impossibile svuotare il writer SYSPRINT. Controlla lo stato del file system.	

Chiave	Gravità	Testo	Dettagli aggiuntivi
BA-U7070	Errore	DSNTIAD: inizializzazione del lettore SYSIN non riuscita. Controlla l'istruzione SYSIN DD.	
BA-U7080	Errore	DSNTIAD: Impossibile chiudere lo scrittore SYSPRINT. Controlla lo stato del file system.	
BA-U7090	Warn	DSNTIAD: conversione SQL non riuscita, utilizzando il codice SQL originale. Rivedi la sintassi SQL per verificare la compatibilità con il database di destinazione.	
BA-U7100	Warn	DSNTIAD: impossibile rilevare il tipo di database, il valore predefinito è PostgreSQL. Verifica la connessione al database e la configurazione del driver JDBC.	
BA-U7110	Warn	DSNTIAD: istruzione SELECT rifiutata . DSNTIAD non supporta le istruzioni SELECT.	

AWS Transform for mainframe Runtime APIs

Il AWS Transform for mainframe Runtime utilizza diverse applicazioni Web per esporre gli endpoint REST, fornendo modi per interagire con le applicazioni modernizzate utilizzando client REST (ad esempio chiamando 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

- Endpoint disponibili per l'utente durante la creazione URLs
- Endpoint per l'applicazione Gapwalk in AWS Transform for mainframe
- Blusamendpoint REST della console delle applicazioni
- Gestisci la console dell'applicazione JICS in AWS Transform for mainframe
- Strutture di dati per AWS Transform per utenti mainframe

Endpoint disponibili per l'utente durante la creazione URLs

Questo argomento elenca URLs i percorsi root per gli endpoint. 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.

Endpoint per l'applicazione Gapwalk in AWS Transform for mainframe

In questo argomento, scopri gli endpoint per l'applicazione web Gapwalk. Questi utilizzano il percorso principale. `/gapwalk-application`

Argomenti

- Endpoint correlati ai job Batch (modernizzati JCLs e simili)
- Endpoint delle metriche
- Altri endpoint
- Endpoint relativi alle code di lavoro

Endpoint correlati ai job Batch (modernizzati JCLs 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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:

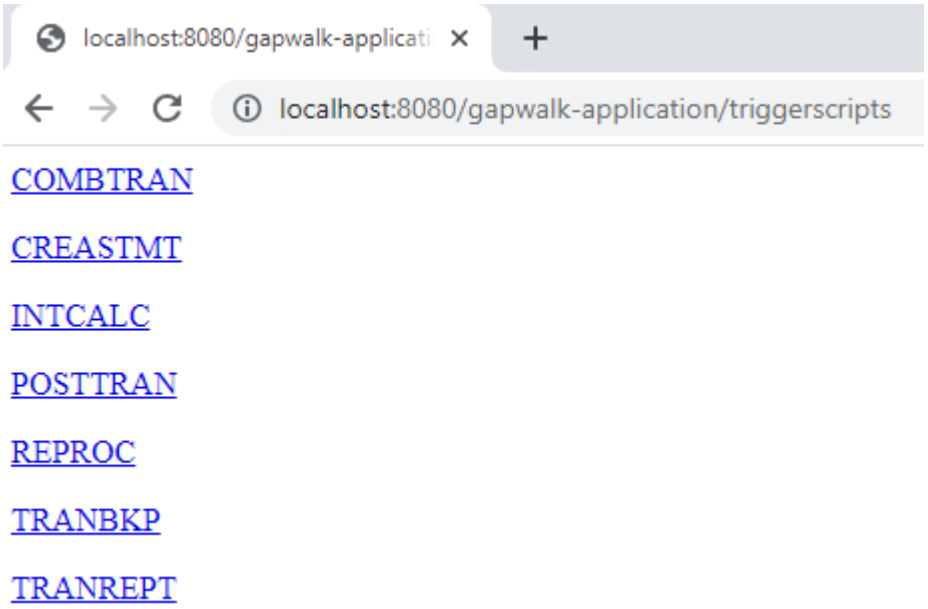
```
<p><a href=./script/COMBTRAN>COMBTRAN</a></p><p><a href=./script/CREASTMT>CREASTMT</
a></p><p><a href=./script/INTCALC>INTCALC</a></p><p><a href=./script/
POSTTRAN>POSTTRAN</a></p><p><a href=./script/REPROC>REPROC</a></p><p><a href=./script/
TRANBKP>TRANBKP</a></p><p><a href=./script/TRANREPT>TRANREPT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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.



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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/script/{scriptId:.+}`
- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/post/script/{scriptId:.+}`
- Argomenti:
  - identificatore dello script per avviare la convalida dell'input: l'ID dello script non deve essere vuoto, non può superare i 255 caratteri e deve corrispondere allo schema: `^[a-zA-Z0-9._-]+$`
  - opzionalmente: parametri da passare allo script, utilizzando i parametri di richiesta (visti come a). `Map<String,String>` I parametri forniti verranno aggiunti automaticamente ai collegamenti dello script groovy invocato. Convalida dell'input: la mappa dei parametri non può superare le 50 voci.

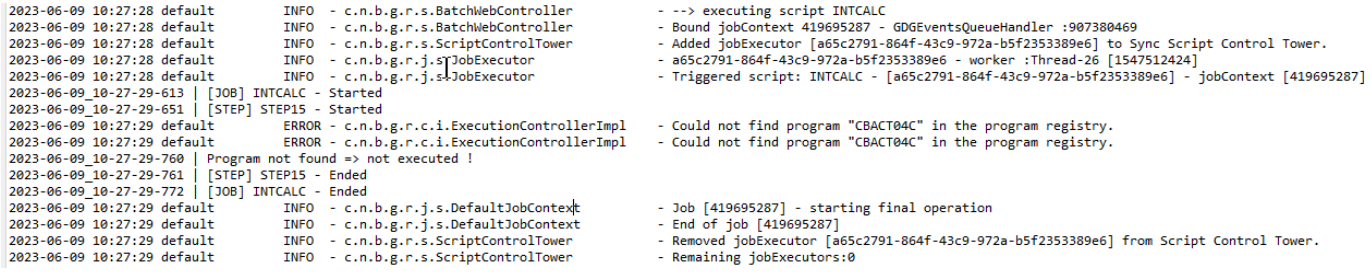
- La chiamata esegue lo script (identificato da scriptID) con parametri opzionali e attende il completamento prima di restituire uno con: *`ResponseEntityString`*
  - HTTP 200: «Fatto». o messaggio di successo JSON in caso di esecuzione riuscita
  - HTTP 200: un messaggio di errore JSON con dettagli sull'errore di esecuzione. Informazioni aggiuntive disponibili nei log del server.

> **Note**
>
> Runtime ora supporta la restituzione del codice di stato HTTP 500 per le esecuzioni di job non riuscite. Vedi Proprietà disponibili per l'applicazione principale per configurare questo codice di risposta.

- Convalida dell'input: l'ID o i parametri dello script non validi restituiranno HTTP 400 Bad Request con i dettagli dell'errore di convalida.

```
{
    "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, il che rende impossibile 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.j.s.JobExecutor               - a65c2791-864f-43c9-972a-b5f2353389e6 - worker :Thread-26 [1547512424]
2023-06-09 10:27:28 default        INFO  - c.n.b.g.r.j.s.JobExecutor               - Triggered script: INTCALC - [a65c2791-864f-43c9-972a-b5f2353389e6] - jobContext [419695287]
2023-06-09_10-27-29-613 | [JOB] INTCALC - Started
2023-06-09_10-27-29-651 | [STEP] STEP15 - 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/triggerscript/{scriptId:.+}`
- Argomenti:
  - identificatore dello script per avviare la convalida dell'input: l'ID dello script non deve essere vuoto, non può superare i 255 caratteri e deve corrispondere allo schema: `^[a-zA-Z0-9._-]+$`
  - opzionalmente: parametri da passare allo script, utilizzando i parametri di richiesta (visti come a). `Map<String,String>` I parametri forniti verranno aggiunti automaticamente ai collegamenti dello script groovy invocato. Convalida dell'input: la mappa dei parametri non può superare le 50 voci.
- 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 l'`settriggerthreadlimit`endpoint 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/POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorsi:, `/triggeredscripts/{status:.+}` `/triggeredscripts/{status:.+}/{namefilter}`
- Argomenti:
  - Status (obbligatorio): lo stato degli script attivati da recuperare. Convalida dell'input: lo stato non deve essere vuoto e non può superare i 50 caratteri. 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 solo i dettagli dei 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. Convalida dell'input: non può superare i 255 caratteri
- Restituisce una raccolta di dettagli sulle esecuzioni dei lavori in formato JSON. Per ulteriori informazioni, consulta Dettagli sull'esecuzione del lavoro, struttura dei messaggi.

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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/getjobexecutioninfo/{jobexecutionid:.+}`
- Argomenti:
    - jobexecutionid (obbligatorio): l'identificatore univoco di esecuzione del lavoro per recuperare i dettagli di esecuzione del lavoro corrispondenti. Convalida dell'input: l'ID di esecuzione del lavoro non deve essere vuoto e non può superare i 255 caratteri
- Restituisce una stringa JSON che rappresenta i dettagli di un singolo processo (vediDettagli sull'esecuzione del lavoro, struttura dei messaggi) 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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'`/kill`endpoint di seguito).

Uccidere una determinata esecuzione di lavoro

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- 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 annullata con la forza. Convalida dell'input: l'identificatore non deve essere vuoto e non può superare i 255 caratteri
- Restituisce un messaggio di testo 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 di AWS Transform for mainframe cercherà di ridurre al minimo l'impatto aziendale dell'interruzione del processo.
- Interrompere forzatamente 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 riservato 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 precedenti al checkpoint).

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- Percorso: `/restarts/{scriptId}/{jobId}`
- Argomenti:
  - scriptId (opzionale - stringa): lo script che viene riavviato.
  - joBid (opzionale - stringa): l'identificatore univoco dell'esecuzione di un lavoro.
- Restituisce un elenco in formato JSON di punti di riavvio esistenti, che può essere utilizzato per riavviare un processo la cui esecuzione non è avvenuta e non è terminata correttamente o per attivare un riavvio ritardato ignorando i passaggi eseguiti in precedenza. Se nessuno script ha registrato alcun checkpoint, il contenuto della pagina sarà «Nessun checkpoint registrato».

Riavvio di un lavoro (in modo sincrono)

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- Percorso: `/restart/{hashcode}/{scriptId}/{skipflag}`
- Argomenti:
  - hashcode (intero - obbligatorio): riavvia l'esecuzione più recente di un job, utilizzando l'hashcode fornito come valore di checkpoint (vedi l'`/restarts`endpoint sopra riportato per scoprire come recuperare un valore di checkpoint valido).
  - scriptID (opzionale - stringa): lo script che viene riavviato.
  - skipflag (opzionale - boolean): salta l'esecuzione del passaggio selezionato (checkpoint) ed emette un riavvio dal passaggio successivo immediato (se presente).
- Restituzioni: vedi la descrizione del reso sopra. `/script`

Riavvio di un lavoro (in modo asincrono)

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/triggerrestart/{hashcode}/{scriptId}/{skipflag}`
- Argomenti:
  - hashcode (intero - obbligatorio): riavvia l'esecuzione più recente di un job, utilizzando l'hashcode fornito come valore di checkpoint (vedi l'`/restarts`endpoint sopra riportato per scoprire come recuperare un valore di checkpoint valido).
  - scriptID (opzionale - stringa): lo script che viene riavviato.
  - skipflag (opzionale - boolean): salta l'esecuzione del passaggio selezionato (checkpoint) ed emette un riavvio dal passaggio successivo immediato (se presente).
- 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 CPUs, memoria disponibile, ecc...). Per impostazione predefinita, il limite di thread è impostato su 5 thread.

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- Percorso: `/settriggerthreadlimit/{threadlimit:.+}`
- Argomento (numero intero): il nuovo limite di thread da applicare. Convalida dell'input: deve essere compreso tra 1 e 1000 inclusi.
- Restituisce un messaggio (`String`) che indica il nuovo limite del thread e quello precedente, o 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 lavori attivate attualmente in esecuzione

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/countrunningtriggeredscripts`
- 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 sull'esecuzione 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: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- Percorso: `/purgejobinformation/{age:.+}`
- Argomenti: un valore intero strettamente positivo che rappresenta l'età in ore delle informazioni da eliminare. Convalida dell'input: deve essere compreso tra 0 e 365 inclusi.
- 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

Endpoint delle metriche

Convalida dell'input: tutti gli endpoint delle metriche convalidano i parametri della richiesta e restituiscono HTTP 400 Bad Request per i valori non validi.

JVM

Questo endpoint restituisce le metriche disponibili relative alla JVM.

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/metrics/batch`
- Argomenti:
  - startTimestamp (opzionale, numero): timestamp iniziale per il filtraggio dei dati. Convalida dell'input: deve essere un valore numerico valido.
  - EndTimestamp (opzionale, numero): timestamp di fine per il filtraggio dei dati. Convalida dell'input: deve essere un valore numerico valido.
  - pagina (opzionale, numero): numero di pagina per l'impaginazione. Convalida dell'input: deve essere un numero intero positivo.
  - PageSize (opzionale, numero): numero di elementi per pagina nell'impaginazione. Convalida dell'input: deve essere un numero intero strettamente positivo, massimo 500.
  - Convalida dell'input: la mappa dei parametri non può superare le 20 voci
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/metrics/transaction`
- Argomenti:
  - startTimestamp (opzionale, numero): timestamp iniziale per il filtraggio dei dati. Convalida dell'input: deve essere un valore numerico valido.
  - EndTimestamp (opzionale, numero): timestamp di fine per il filtraggio dei dati. Convalida dell'input: deve essere un valore numerico valido.
  - pagina (opzionale, numero): numero di pagina per l'impaginazione. Convalida dell'input: deve essere un numero intero positivo.
  - PageSize (opzionale, numero): numero di elementi per pagina nell'impaginazione. Convalida dell'input: deve essere un numero intero strettamente positivo, massimo 500.
  - Convalida dell'input: la mappa dei parametri non può superare le 20 voci
- 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)
- Elenca le sessioni attive

Elencare i programmi registrati

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 alle z/OS utilità potrebbero non essere disponibili per gli asset iSeries modernizzati, in quanto non pertinenti.

Elenco dei servizi registrati

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/services`

- Restituisce l'elenco dei servizi di runtime registrati, come pagina html. I servizi forniti sono forniti dal runtime AWS Transform for mainframe 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 preesistenti) rientrano in questa categoria.

Risposta di esempio:

```
<p>BluesamESDSFileLoader</p><p>BluesamKSDSFileLoader</p><p>BluesamRRDSFileLoader</p>
```

Health status (Stato di integrità)

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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: POST

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER

- 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. Convalida dell'input: XID non deve essere vuoto e non può superare i 255 caratteri.
- Restituisce una serializzazione `com.netfective.bluage.gapwalk.rt.shared.web.TransactionResultBean` JSON, che rappresenta il risultato dell'avvio della transazione JICS.
- Convalida dell'input: i valori XID non validi (vuoti o superiori a 255 caratteri) restituiranno HTTP 400 Bad Request con i dettagli dell'errore di convalida.

Per ulteriori informazioni sui dettagli della struttura, vedere. Struttura dei risultati del lancio della transazione

Avviare una transazione JICS (alternativa)

- metodi supportati: POST

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER

- 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. Convalida dell'input: XID non deve essere vuoto e non può superare i 255 caratteri.
- 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
- Convalida dell'input: i valori XID non validi (vuoti o superiori a 255 caratteri) restituiranno HTTP 400 Bad Request con i dettagli dell'errore di convalida.

Elenca le sessioni attive

- metodi supportati: GET

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN

- percorso: `/activesessionlist`
- Argomenti: nessuno
- Convalida dell'input: la mappa dei parametri non può superare le 20 voci
- Restituisce un elenco di serializzazioni `com.netfective.bluage.gapwalk.application.web.sessiontracker.SessionTrackerObje` in JSON, che rappresenta l'elenco delle sessioni utente attive. Quando il tracciamento delle sessioni è disabilitato, verrà restituito un elenco vuoto.

Endpoint relativi alle code di lavoro

Le code di lavoro sono il supporto di AWS Transform for mainframe per il meccanismo di invio dei AS400 lavori. Le code di lavoro vengono utilizzate AS400 per eseguire lavori 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'offerta di lavoro per il lancio
- Elenca tutti i lavori inviati
- Rilasciare 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
- Invia un lavoro con un programma ripetuto
- Elenca tutti i lavori ripetuti inviati

- Annulla la pianificazione di un lavoro ripetuto

Elenca le code disponibili

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN
- Percorso: `/restart/{name}`
- Argomento: il nome della coda da avviare/riavviare, come stringa - obbligatorio. Convalida dell'input: il nome della coda non deve essere vuoto e non può superare i 255 caratteri.
- L'endpoint non restituisce nulla ma si affida piuttosto allo stato http per indicare l'esito dell'operazione: start/restart

HTTP 200

l' start/restart operazione 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 start/restart tentativo (i log del server devono essere ispezionati per capire cosa è andato storto).

Invia un'offerta di lavoro per il lancio

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/submit`
- Argomento: obbligatorio come corpo della richiesta, serializzazione JSON di un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` oggetto. Per ulteriori informazioni, consulta Invia il lavoro e pianifica l'input del 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 inviati

- Metodo supportato: GET

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/list-jobs?status={status}&size={size}&page={page}&sort={sort}`
- Argomenti:
  - pagina: numero di pagina da recuperare (impostazione predefinita = 1)
  - dimensione: dimensione della pagina (impostazione predefinita = 50, max = 300)
  - sort: L'ordine dei lavori. (predefinito = «ExecutionID»). «ExecutionID» è attualmente l'unico valore supportato

  - status: (opzionale) Se presente, filtrerà in base allo stato.
- Restituisce un elenco di tutti i lavori pianificati, come stringa JSON. Per un esempio di risposta, vedereElenco delle risposte ai lavori pianificati.

Rilasciare tutti i lavori che sono «in attesa»

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/release-all`
- Restituisce un messaggio che indica l'esito dell'operazione di tentativo di rilascio. Ecco due casi possibili:
  - HTTP 200 e un messaggio «Tutti i lavori sono stati rilasciati con successo!» se tutti i lavori sono stati rilasciati con successo.
  - HTTP 503 e un messaggio «Jobs not released». Si è verificato un errore sconosciuto. Vedi «log» 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

Per un determinato nome di lavoro, è possibile inviare più offerte di lavoro, con numeri di lavoro diversi (l'unicità di un lavoro è garantita da una coppia <job name, job number>). L'endpoint tenterà di rilasciare tutte le offerte di lavoro con il nome del lavoro specificato, che sono «in attesa».

- Metodo supportato: POST

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/release/{name}`
- Argomenti: il nome del lavoro da cercare, sotto forma di stringa. Obbligatorio.
- Restituisce un messaggio che indica il risultato dell'operazione di tentativo di rilascio. Ecco 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

  Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/release/{name}/{number}`
- Argomenti:

  nome

  il nome del lavoro da cercare, come stringa. Obbligatorio.

  numero

  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.
  - <name/number>HTTP 503 e un messaggio «Job>Non rilasciato. Si è verificato un errore sconosciuto. Vedi il registro per maggiori dettagli» se qualcosa è andato storto con il tentativo di rilascio.

Invia un lavoro con un programma ripetuto

Pianifica un lavoro che verrà eseguito con una pianificazione ripetuta.

- Metodo supportato: POST

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER

- Percorso: `/schedule`
- Argomento: il corpo della richiesta deve contenere una serializzazione JSON di un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` oggetto.

Elenca tutti i lavori ripetuti inviati

- Metodo supportato: GET

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN

- Percorso: `/schedule/list?status={status}&size={size}&page={page}&sort={sort}`
- Argomenti:
  1. pagina: numero di pagina da recuperare (impostazione predefinita = 1)
  2. dimensione: dimensione della pagina (impostazione predefinita = 50, max = 300)
  3. sort: L'ordine dei lavori. (predefinito = «id»). «id» è l'unico valore supportato per ora.
  4. status: (opzionale) Se presente, filtrerà in base allo stato. I valori possibili sono quelli indicati nella sezione 1.
  5. status: (opzionale) Se presente, filtrerà in base allo stato. I valori possibili sono quelli indicati nella sezione 1.
  6. Restituisce un elenco di tutti i lavori pianificati, come stringa JSON.

Annulla la pianificazione di un lavoro ripetuto

Rimuove un lavoro creato in base a una pianificazione ripetuta. Lo stato di pianificazione dei processi è impostato su INATTIVO.

- Metodo supportato: POST

Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN

- Percorso: `/schedule/remove/{schedule_id}`
- Argomento:`schedule_id`, l'identificatore del lavoro pianificato da rimuovere.

Blusamendpoint REST della console delle applicazioni

In questa sezione, puoi conoscere la console delle Blusam applicazioni, un'API progettata per semplificare la gestione dei set di dati VSAM modernizzati. Gli endpoint per l'applicazione Blusam Web utilizzano il percorso principale. `/bac`

Argomenti

- Set di dati (endpoint correlati)
- Set di dati in blocco: endpoint correlati
- Registri
- Maschere
- Altro
- Endpoint BAC per la gestione degli 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 (POST)
- Carica un set di dati (GET)
- 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

È possibile utilizzare questo endpoint per creare una definizione di set di dati.

- Metodi supportati: POST

- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/createDataSet`
- Argomenti:

  nome

  (obbligatorio, stringa): il nome del set di dati.

  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

È possibile utilizzare questo endpoint per caricare file sul server. Il file viene archiviato in una cartella temporanea che corrisponde a ciascun utente specifico. Usa questo endpoint ogni volta che devi caricare un file.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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 (POST)

Dopo aver creato `createDataSet` la definizione del set di dati, è possibile caricare i record associati al file caricato in un set di dati specifico.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.

- 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.

Carica un set di dati (GET)

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/loadDataSet`
- Argomenti:

  listcatFileOrDatasetName

  (obbligatorio, stringa): il nome del set di dati.

  DataSetFile

  (obbligatorio, stringa): il nome del file 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
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/loadDataSetFromS3`
- Argomenti:

  Elenca la posizione dei file CAT 3

  (obbligatorio, stringa): la posizione Amazon S3 del file listcat.

  Posizione di DataSetFiles3

  (obbligatorio, stringa): la posizione Amazon S3 del file del set di dati.

region

(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
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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.

  region

  (obbligatorio, stringa): il Regione AWS bucket Amazon S3.

  kmsKeyId

  (opzionale, stringa): l' AWS KMS ID da utilizzare per la crittografia del set di dati esportato nel 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.

- Metodi supportati: POST, GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/clearDataSet`
- Argomenti:

  nome

  (obbligatorio, stringa): il nome del set di dati da cancellare. Quando si utilizza il metodo GET, il nome del parametro è`datasetName`.
- Restituisce lo stato della richiesta.

Eliminare un set di dati

Elimina la definizione e i record del set di dati.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.

- 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 (GET)
- Esporta set di dati (POST)
- Crea più set di dati
- Elenca tutti i set di dati
- Elenca direttamente tutti i set di dati
- Elenca direttamente tutti i set di dati per pagina
- Set di dati di streaming
- 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
- Ottieni un set di dati
- Carica listcat dal file JSON

Esporta set di dati (GET)

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/exportDataSet`
- Argomenti:

  datasetName

  (obbligatorio, stringa): il nome del set di dati da esportare.

datasetOutputFile

(obbligatorio, stringa): il percorso della cartella in cui si desidera archiviare il set di dati esportato sul server.

rdw

(obbligatorio, booleano): se si desidera che la parola descrittiva del record (RDW) faccia parte dei record esportati. Se il set di dati contiene record a lunghezza fissa, il valore di questo parametro viene ignorato.

- Restituisce lo stato della richiesta e il percorso del file contenente il set di dati esportato (se presente). Se il set di dati è nullo nella risposta, significa che il sistema non è stato in grado di individuare un set di dati con il nome specificato.

Esporta set di dati (POST)

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/exportDataSet`
- Argomenti:

  Parametri di dump

  (obbligatorio, BACRead Parametri): parametri di lettura Bluesam.
- Restituisce lo stato del set di dati esportato.

Crea più set di dati

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/createAllDataSets`
- Argomenti:
  - Elenco dei set di dati

    nome

    (obbligatorio, stringa): il nome del set di dati.

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
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/directListDataSet`
- Argomenti: Nessuno
- Restituisce: lo stato della richiesta e l'elenco dei set di dati.

Elenca direttamente tutti i set di dati per pagina

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/directListDataSetByPage`
- Argomenti:

  nome

  (obbligatorio, stringa): il nome del set di dati. Il valore predefinito è `%` (tutti i set di dati) se non specificato.

  page

  (obbligatorio, int): il numero di pagina (minimo 0).

  pageSize

  (obbligatorio, int): la dimensione della pagina (minimo 1, massimo 500).
- Restituisce: lo stato della richiesta e l'elenco dei set di dati.

Set di dati di streaming

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/streamDataset`
- Argomenti:

  datasetName

  (obbligatorio, stringa): il nome del set di dati.
- Restituisce: un flusso dei set di dati richiesti.

Elimina tutti i set di dati

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/getDataSetsDefinitionFromUploadedListcat`
- Argomenti: Nessuno
- Restituisce: un elenco di set di dati

Ottieni un set di dati

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/getDataSet`
- Argomenti:

  nome

  (obbligatorio, stringa): il nome del set di dati.
- Restituisce il set di dati richiesto.

Carica listcat dal file JSON

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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
- Convalida un record
- Ottieni un albero dei record

Creazione di un record

È possibile utilizzare questo endpoint per creare un nuovo record.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

È possibile utilizzare questo endpoint per leggere un set di dati.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

È possibile utilizzare questo endpoint per eliminare un record da un set di dati.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

È possibile utilizzare questo endpoint per aggiornare un record associato a un set di dati.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

  maschera

  (opzionale, Maschera): l'oggetto maschera da applicare durante l'aggiornamento.
- Restituisce lo stato della richiesta e il set di dati con i record.

Salva un record

È possibile utilizzare questo endpoint per salvare un record in un set di dati e utilizzare una maschera.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

  maschera

  (opzionale, Maschera): l'oggetto maschera da applicare durante il salvataggio.

- Restituisce lo stato della richiesta e il set di dati con i record.

Convalida un record

Usa questo endpoint per convalidare un record.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/crud/validateRecord`
- Argomenti:

  set di dati

  (obbligatorio, DataSet): l'oggetto del set di dati

  record

  (opzionale, Record): il record da convalidare.

  maschera

  (opzionale, Mask): l'oggetto maschera da applicare durante la convalida.
- Restituisce lo stato della richiesta e il set di dati con i record.

Ottieni un albero dei record

Usa questo endpoint per ottenere l'albero gerarchico di un record.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/crud/getRecordTree`
- Argomenti:

  set di dati

  (obbligatorio, DataSet): l'oggetto del set di dati

  record

  (richiesto, Record): il record da recuperare

maschera

(opzionale, Maschera): l'oggetto maschera.

- Restituisce lo stato della richiesta e l'albero gerarchico del record richiesto.

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

È possibile utilizzare questo endpoint per recuperare tutte le maschere associate a un set di dati specifico.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/crud/loadMasks`
- Variabili del percorso:

  RecordSize:... /loadMasks/ {recordSize}

  (opzionale, numerico): la dimensione del record, filtra le maschere caricate che corrispondono a questa dimensione del record
- 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

È possibile utilizzare questo endpoint per applicare una maschera a un set di dati specifico.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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

È possibile utilizzare questo endpoint per applicare una maschera e un filtro a un set di dati specifico.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/crud/applyMaskFilter`
- Argomenti:

  set di dati

  (obbligatorio, DataSet): l'oggetto del set di dati

  maschera

  (obbligatorio, Maschera): l'oggetto maschera

  filter

  (obbligatorio, Filter): l'oggetto filtro da applicare.
- 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 configurazione multi-schema di PostgreSQL

Controlla la cache di riscaldamento

Verifica se la cache di riscaldamento è abilitata per un set di dati specifico.

- Metodi supportati: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- 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.

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/isEnableCache`
- Argomenti: Nessuno
- Restituisce true se la memorizzazione nella cache è abilitata.

Abilita la cache

- Metodi supportati: POST
- Richiede l'autenticazione e i ruoli ROLE_ADMIN e ROLE_SUPER_ADMIN.
- Percorso: `/api/services/rest/bluesamservice/enableDisableCache/{enable}`
- Argomenti:

  abilita

  (obbligatorio, booleano): se impostato su true, abiliterà la memorizzazione nella cache.

  set di dati

  (obbligatorio, DataSet): l'oggetto del set di dati.
- Restituisce Nessuno

Controlla la cache RAM allocata

È possibile utilizzare questo endpoint per recuperare la memoria cache RAM allocata.

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/allocatedRamCache`
- Argomenti: Nessuno
- Restituisce: la dimensione della memoria come stringa

Controlla la persistenza

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/persistence`
- Argomenti: Nessuno
- Restituisce: la persistenza usata come stringa

Controlla i tipi di set di dati supportati

- Metodi supportati: GET

- Percorso: `/api/services/rest/bluesamservice/getDataSetTypes`
- Richiede l'autenticazione e il ruolo ROLE_USER.
- 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/bluesamserver/serverIsUp`
- Argomenti: Nessuno
- Restituzioni: nessuna. Il codice di stato della risposta HTTP 200 indica che il server è attivo e funzionante.

Controlla la configurazione multi-schema di PostgreSQL

Verifica se la configurazione multi-schema di PostgreSQL è abilitata.

- Metodi supportati: GET
- Richiede l'autenticazione e il ruolo ROLE_USER.
- Percorso: `/api/services/rest/bluesamservice/isPostgresMultiSchema`
- Argomenti: Nessuno
- Restituisce: true se la configurazione multi-schema di PostgreSQL è abilitata e false in caso contrario.

Endpoint BAC per la gestione degli utenti

Utilizza i seguenti endpoint per gestire le interazioni con gli utenti.

Argomenti

- Effettua il login di un utente
- Verifica se nel sistema esiste almeno un utente
- Registra un nuovo utente
- Ottieni informazioni sull'utente
- Elencare gli utenti

- Eliminazione di un utente
- Disconnette l'utente corrente

Effettua il login di un utente

- Metodo supportato: POST
- Percorso: `/api/services/security/servicelogin/login`
- Argomenti: Nessuno
- Restituisce la serializzazione JSON di un `com.netfective.bluage.bac.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": "some-admin",
     "password": null,
     "roles": [
       {
         "id": 0,
         "roleName": "ROLE_ADMIN"
       }
     ]
   }
```

Verifica se nel sistema esiste almeno un utente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogin/hasAccount`
- Argomenti: Nessuno
- Restituisce il valore booleano `true` se è stato creato almeno un utente diverso dall'utente super amministratore predefinito. Restituisce altrimenti`false`.

Registra un nuovo utente

- Metodo supportato: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.

- Percorso: `/api/services/security/servicelogin/recorduser`
- Argomenti: la serializzazione JSON di un `com.netfective.bluage.bac.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 BAC.
- Restituisce il valore booleano `true` se l'utente è stato creato con successo. Restituisce altrimenti`false`.
- Esempio di richiesta JSON:

```
{
    "login": "simpleuser",
    "password": "simplepassword",
    "roles": [
      {
        "id": 2,
        "roleName": "ROLE_USER"
      }
    ]
  }
```

  Di seguito sono riportati i due valori validi per`roleName`:

  - `ROLE_ADMIN`: può gestire Blusam risorse e utenti.
  - `ROLE_USER`: può gestire Blusam le risorse ma non gli utenti.

Ottieni informazioni sull'utente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogin/userInfo`
- Argomenti: Nessuno
- Restituisce il nome utente e il ruolo dell'utente attualmente connesso

Elencare gli utenti

- Metodo supportato: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/security/servicelogin/listusers`

- Argomenti: Nessuno
- Restituisce un elenco di`com.netfective.bluage.bac.entities.SignOn`, serializzato come JSON.

Eliminazione di un utente

> **Important**
>
> Questa operazione non può essere annullata. L'utente eliminato non sarà più in grado di connettersi all'applicazione BAC.

- Metodo supportato: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/security/servicelogin/deleteuser`
- Argomenti: la serializzazione JSON di un `com.netfective.bluage.bac.entities.SignOn` oggetto che rappresenta l'utente da rimuovere dall'archivio.
- Restituisce il valore booleano `true` se l'utente è stato rimosso con successo.

Disconnette l'utente corrente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogout/logout`
- Argomenti: Nessuno
- Restituisce il messaggio JSON `{"success":true}` se l'utente corrente è stato disconnesso con successo. La sessione HTTP correlata verrà invalidata.

Gestisci la console dell'applicazione JICS in AWS Transform for mainframe

Il componente JICS è il supporto AWS Transform for mainframe per la modernizzazione delle risorse CICS legacy. L'applicazione Web della console dell'applicazione JICS 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 della console dell'applicazione JICS utilizzano il percorso root. `/jac/`

Argomenti

- Gestione delle risorse JICS
- Altro
- 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
- Recupera risorse JICS
- Elenca JICS 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 che utilizza 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
- Eliminare un TSMODEL
- Elimina elementi
- Crea una LISTA
- Creare un GRUPPO
- Considerazioni comuni sulla creazione di RISORSE
- Crea una TRANSAZIONE

- Crea un PROGRAMMA
- Crea un FILE
- Crea un TDQUEUE
- Crea un TSMODEL
- Crea elementi
- 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
- Aggiorna elementi
- Elementi Upsert
- Recupera elementi
- Operazione JICS CRUD

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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/listJicsListsAndGroups`
- Argomenti: Nessuno
- 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
    }
  ]
```

Recupera risorse JICS

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/retrieveJicsResources`
- Argomenti: un payload JSON che rappresenta le risorse JICS che si desidera recuperare. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.request.RetrieveOperationRequest`
- Restituisce: un elenco di oggetti JicsResource serializzati. Gli oggetti non vengono restituiti in un ordine particolare e sono di diversi tipi, come PROGRAM, TRANSACTION, FILE e così via.

Elenca JICS GROUPS

- Metodo supportato: GET
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/listJicsGroups`
- Argomenti: Nessuno
- 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/listGroupsForList`
- Argomenti: un payload JSON, che rappresenta la LISTA JICS di cui GROUPS stai cercando. Questa è la serializzazione JSON di un oggetto. `com.netfective.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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/listResourcesForGroup`
- Argomenti: un payload JSON, che rappresenta il JICS GROUP di cui state 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 non vengono restituiti in un ordine particolare e sono di diversi tipi, come PROGRAM, TRANSACTION, FILE e così via.

ELENCA le risorse JICS per un determinato GRUPPO (alternativa che utilizza un nome)

- Metodo supportato: POST
- Richiede l'autenticazione
- Percorso: `/api/services/rest/jicsservice/listResourcesForGroupName`
- Argomenti: il nome del GRUPPO che possiede le risorse che stai 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, come PROGRAM, TRANSACTION, FILE e così via.

Modifica dei GRUPPI proprietari di diverse LISTE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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, si «aggiunge» il gruppo denominato «TEST» alla LISTA denominata «MURACHS».

- Restituisce un valore booleano. Se il valore è 'true', le modifiche di LISTS sono state mantenute correttamente nell'archivio JICS sottostante.

Eliminare un ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 sull'archivio JICS sottostante.

Eliminare un GRUPPO

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 con successo sullo storage JICS sottostante.

Eliminare una TRANSAZIONE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.netfective.bluage.jac.entities.JACTransaction`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione di TRANSACTION è stata eseguita correttamente sullo storage JICS sottostante.

Eliminare un PROGRAMMA

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.netfective.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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.netfective.bluage.jac.entities.JACFile`
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione del FILE è stata eseguita correttamente sulla memoria JICS sottostante.

Elimina un TDQUEUE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/deleteTDQueue`
- Argomenti: un payload JSON, che rappresenta il JICS TDQUEUE da eliminare. Questa è la serializzazione JSON di un file `com.netfective.bluage.jac.entities. JACTDQueue`oggetto.
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione di TDQUEUE è stata eseguita correttamente sullo storage JICS sottostante.

Eliminare un TSMODEL

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/deleteTSModel`
- Argomenti: un payload JSON, che rappresenta il JICS TSMODEL da eliminare. Questa è la serializzazione JSON di un file `com.netfective.bluage.jac.entities. JACTSModel`oggetto.
- Restituisce un valore booleano. Se il valore è 'true', l'eliminazione TSMODEL è stata eseguita correttamente sull'archivio JICS sottostante.

Elimina elementi

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/deleteElements`
- Argomenti: un payload JSON che rappresenta gli elementi JICS da eliminare.
- Restituisce un valore booleano che `true` indica che l'eliminazione è stata eseguita correttamente nell'archivio JICS sottostante.

Crea una LISTA

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/createList`
- Argomenti: un payload JSON, che rappresenta la LISTA JICS da creare. Questa è la serializzazione JSON di un file `com.netfective.bluage.jac.entities. JACList`oggetto.
- Restituisce un valore booleano. Se il valore è 'true', l'ELENCO è stato creato correttamente nell'archivio JICS sottostante.

> **Note**
>
> L'ELENCO verrà sempre creato vuoto. Allegare GROUPS alla LIST richiederà un'altra operazione.

Creare un GRUPPO

- Metodo supportato: POST
- Richiede l'autenticazione e i seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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. `groupName`Deve 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: si tenta di creare la transazione «ARIT» nel gruppo «TEST», ma in quel gruppo esiste già una transazione con quel nome.

```
{
    "jacType":"JACTransaction",
    "name":"ARIT",
    "groupName":"TEST",
    "isActive":true
  }
```

Riceverai 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.netfective.bluage.jac.server.services.rest.impl.JicsServiceImpl.createElement(JicsServiceI
```

Crea una TRANSAZIONE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.netfective.bluage.jac.entities.JACTransaction`
- Restituisce un valore booleano. Se il valore è 'true', la TRANSACTION è stata creata correttamente nell'archivio JICS sottostante.

Crea un PROGRAMMA

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 con successo nella memoria JICS sottostante.

Crea un FILE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 con successo nella memoria JICS sottostante.

Crea un TSMODEL

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 dove `true` indica che la creazione di elementi è stata eseguita con successo nella memoria JICS sottostante.

Crea elementi

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/createElements`
- Argomenti: un payload JSON che rappresenta gli elementi JICS da creare.
- Restituisce un valore booleano. Se il valore è 'true', gli elementi sono stati creati correttamente nella memoria JICS sottostante.

Aggiorna un ELENCO

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 terrà conto dei bambini.
- 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 propagherà 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 JICS sottostante (in caso contrario, si riceverà una risposta BAD REQUEST (HTTP STATUS 400) dall'endpoint).

Aggiorna una TRANSAZIONE

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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 e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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', è TDQueue stato aggiornato con successo nella memoria JICS sottostante.

Aggiorna un TSMODEL

- Metodo supportato: POST

- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- 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.

Aggiorna elementi

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/updateElements`
- Argomenti: un payload JSON che rappresenta gli elementi da aggiornare.
- Restituisce un valore booleano che `true` indica che l'aggiornamento degli elementi è stato eseguito correttamente nell'archivio JICS sottostante.

Elementi Upsert

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/upsertElements`
- Argomenti: un payload JSON che rappresenta gli elementi da sconvolgere.
- Restituisce un valore booleano dove `true` indica che gli elementi upsert sono stati utilizzati con successo nella memoria JICS sottostante.

Recupera elementi

- Metodo supportato: GET
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER

- Percorso: `/api/services/rest/jicsservice/retrieveElements`
- Argomenti: Nessuno
- Restituisce un elenco di tutte le risorse JICS serializzate.

Operazione JICS CRUD

- Metodo supportato: POST
- Richiede l'autenticazione e uno dei seguenti ruoli: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Percorso: `/api/services/rest/jicsservice/jicsCrudOperation`
- Argomenti: un payload JSON che rappresenta le risorse JICS che stai cercando. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.request.JicsCrudOperationRequest`
- Restituisce un payload JSON che rappresenta la risposta. Questa è la serializzazione JSON di un oggetto. `com.netfective.bluage.jac.entities.request.JicsCrudOperationResponse`

Altro

Argomenti

- Stato di salute del server JICS

Stato di salute del server JICS

- Metodo supportato: GET
- Percorso: `/api/services/rest/jicsserver/serverIsUp`
- Argomenti: Nessuno
- Restituzioni: nessuna. Una risposta HTTP STATUS 200 indica che il server è attivo e funzionante.

Endpoint di gestione degli utenti JAC

Utilizza i seguenti endpoint per gestire le interazioni con gli utenti.

Argomenti

- Registrazione di un utente
- Verifica se nel sistema esiste almeno un utente
- Registrazione di un nuovo utente
- Informazioni sull'utente
- Elencare gli utenti
- Eliminazione di un utente
- Disconnetti l'utente corrente

Registrazione di un utente

- Metodo supportato: POST
- Percorso: `/api/services/security/servicelogin/login`
- Argomenti: Nessuno
- Restituisce la serializzazione JSON di un `com.netfective.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": "some-admin",
    "password": null,
    "roles": [
      {
        "id": 0,
        "roleName": "ROLE_ADMIN"
      }
    ]
  }
```

Verifica se nel sistema esiste almeno un utente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogin/hasAccount`
- Argomenti: Nessuno

- Restituisce il valore booleano `true` se è stato creato almeno un utente diverso dall'utente super amministratore predefinito. Restituisce altrimenti`false`.

Registrazione di un nuovo utente

- Metodo supportato: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/security/servicelogin/recorduser`
- Argomenti: la serializzazione JSON di un `com.netfective.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.
- Restituisce il valore booleano `true` se l'utente è stato creato con successo. Restituisce altrimenti`false`.

Richiesta di esempio:

```
{
    "login": "simpleuser",
    "password": "simplepassword",
    "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.

Informazioni sull'utente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogin/userInfo`
- Argomenti: Nessuno

- Restituisce il nome utente e i ruoli dell'utente attualmente connesso.

Elencare gli utenti

- Metodo supportato: GET
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/security/servicelogin/listusers`
- Argomenti: Nessuno
- Restituisce un elenco di`com.netfective.bluage.jac.entities.SignOn`, serializzato come JSON.

Eliminazione di un utente

- Metodo supportato: POST
- Richiede l'autenticazione e il ruolo ROLE_ADMIN.
- Percorso: `/api/services/security/servicelogin/deleteuser`
- Argomenti: la serializzazione JSON di un `com.netfective.bluage.jac.entities.SignOn` oggetto che rappresenta l'utente da rimuovere dall'archivio.
- Restituisce il valore booleano `true` se l'utente è stato rimosso con successo.

> **Important**
>
> Questa operazione non può essere annullata. L'utente eliminato non sarà in grado di connettersi nuovamente all'applicazione JAC.

Disconnetti l'utente corrente

- Metodo supportato: GET
- Percorso: `/api/services/security/servicelogout/logout`
- Argomenti: Nessuno
- Restituisce il messaggio JSON `{"success":true}` se l'utente corrente è stato disconnesso con successo. La sessione HTTP correlata verrà invalidata.

Strutture di dati per AWS Transform per utenti mainframe

Puoi scoprire varie strutture di dati per il motore AWS Transform for mainframe nella sezione seguente.

Argomenti

- Dettagli sull'esecuzione del lavoro, struttura dei messaggi
- Struttura dei risultati del lancio della transazione
- Struttura dei risultati del record di avvio delle transazioni
- Possibile stato di un lavoro in coda
- Invia il lavoro e pianifica l'input del lavoro
- Elenco delle risposte ai lavori pianificati
- Elenco delle risposte ai lavori ripetuti

Dettagli sull'esecuzione del lavoro, struttura dei messaggi

I dettagli di esecuzione di ogni processo 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.

executionMode

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 andata a buon fine.
- `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

(Facoltativo) 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

(Facoltativo) 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:

Completato

```
{
    "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:

Completato

```
{
    "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 il lavoro e pianifica l'input del lavoro

L'input del processo di invio e pianificazione è la serializzazione JSON di un `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` oggetto. L'input di esempio riportato di seguito mostra tutti i campi relativi a tale bean.

Esempio di input per l'invio del lavoro:

```
{
    "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
}
```

Esempio di input per pianificare un lavoro:

```
{
     "scheduleCron": "*/2 * * * * ?",
     "programName":"LOGPGM",
     "programParams": {
         "cl_sbmjob_param_json": "[\"./output/schedule-job-log.txt\", \"Every 2
 seconds!\"]"
     },
     "localDataAreaValue":"",
     "userName":"PVO",
     "jobName":"LOGGERJOB",
     "jobPriority":5,
     "jobQueue":"queue1",
     "scheduleMisfirePolicy": 4,
     "startTime": "2003/05/04 07:00:00.000 GMT-06:00",
     "endTime": "2003/05/04 07:00:07.000 GMT-06:00"
 }
```

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 sospeso, 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 format MMddyy o dd MMyyyy (la dimensione dell'input determinerà il formato utilizzato)

Orario

Può essere fornito con format HHmm o HHmmss (la dimensione dell'input determinerà il formato utilizzato)

Parametri del programma

Verrà passato al programma come mappa.

scheduleMisfirePolicy

Definisce la strategia utilizzata quando un trigger viene attivato male. Di seguito sono riportati i valori possibili:

1. Rilascia il primo errore e scarta gli altri.
2. Invia un lavoro in sospeso per il primo errore e scarta gli altri errori.
3. Scarta l'errore di accensione.
4. Rilascia tutti i fallimenti. La coda dei lavori eseguirà tutti i lavori.

Elenco delle risposte ai lavori pianificati

Questa è la struttura dell'endpoint list-jobs job queue. Il messaggio di invio del lavoro utilizzato per inviare quel lavoro fa parte della risposta. 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.

```
[
  {
    "jobName": "PTA0044",
    "userName": "USER1",
    "jobNumber": 9,
    "jobPriority": 5,
    "status": "HOLD",
    "jobDelay": 0,
    "startDate": null,
    "endDate": null,
    "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,
      "scheduleCron": null,
      "save": false,
      "scheduleMisfirePolicy": 4,
      "omitdates": null
    },
    "executionId": 1,
    "jobScheduledId": 0,
    "jobScheduledAt": null
  },
  {
    "jobName": "PTA0044",
    "userName": "USER1",
    "jobNumber": 9,
    "jobPriority": 5,
    "status": "COMPLETED",
    "jobDelay": 0,
    "startDate": "2022-10-13T22:48:34.025+00:00",
    "endDate": "2022-10-13T22:52:54.475+00:00",
    "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,
      "scheduleCron": "*/20 * * * * ?",
      "save": false,
      "scheduleMisfirePolicy": 4,
      "omitdates": null
    },
    "executionId": 2,
    "jobScheduledId": 0,
    "jobScheduledAt": null
  }
]
```

Elenco delle risposte ai lavori ripetuti

Questa è la struttura dell'endpoint /schedule/listjob queue.

```
[
  {
    "id": 1,
    "status": "ACTIVE",
    "jobNumber": 1,
    "userName": "PVO",
    "msg": {
      "messageQueueName": null,
      "scheduleDate": null,
      "scheduleTime": null,
      "startTime": "2024/03/07 21:12:00.000 UTC",
      "endTime": "2024/03/07 21:13:59.000 UTC",
      "programName": "LOGPGM",
      "programParams": {"cl_sbmjob_param_json": "[\"./output/schedule-job-log.txt\",
 \"Every 20 seconds!\"]"},
      "localDataAreaValue": "",
      "userName": "PVO",
      "jobName": "LOGGERJOB",
      "jobNumber": 1,
      "jobScheduleId": 1,
      "jobPriority": 5,
      "executionDate": null,
      "jobQueue": "queue1",
      "jobOnHold": false,
      "scheduleCron": "*/20 * * * * ?",
      "save": false,
```

```
      "scheduleMisfirePolicy": 4,
      "omitdates": null
    },
    "lastUpdatedAt": "2024-03-07T21:11:13.282+00:00",
    "lastUpdatedBy": ""
  }
]
```

AWS Transform per mainframe Runtime Utilities

Gli script JCL mainframe utilizzano programmi aziendali e utilità di sistema per i clienti. AWS Transform for mainframe Runtime include il supporto per le utilità di sistema di uso comune per consentire la corretta modernizzazione di JCL su Groovy, oltre a utili utilità aggiuntive.

Lo scopo di questo documento è:

- Spiegare come configurare e distribuire le utilità;
- Elenca le utilità di sistema esistenti supportate e fornisci dettagli sul loro utilizzo e scopo.

Configura e distribuisci l'applicazione web Utility

Le utilità sono fornite in `gapwalk-utility-pgm-<version>.war` (`<version>`dov'è la versione runtime di AWS Transform for mainframe). Implementa applicazioni `gapwalk-application` parallele e modernizzate per consentire l'accesso alle utilità da script JCL modernizzati.

Tutte le utilità supportate sono registrate come programmi nel «Registro dei programmi» (consultate questo sito the section called "Esecuzione e chiamata di programmi" per maggiori dettagli sulla registrazione dei programmi e sul loro utilizzo negli script o in altri programmi). Pertanto, possono essere richiamati da altri programmi o da script di lavoro modernizzati.

L'applicazione web delle utilità ha il proprio file di configurazione, denominato`application-utility-pgm.yml`, il cui contenuto è dettagliato in questo documento. the section called "Configurare l'accesso alle utilità per le applicazioni gestite"

Programmi di utilità disponibili

Programmi di utilità disponibili, elencati in base al rispettivo dominio di applicazione:

- the section called "Utilità per set di dati"

- the section called "BLUESAMCOPY/BLUESAMCREATE/BLUESAMDELETE/BLUESAMCLEAR"
- the section called "BPXWDYN"
- the section called "GDGUTILS"
- the section called "ICEGENER/SYNCGENR"
- the section called "IDCAMS/KQCAMS"
- the section called "IEBGENER/JSDGENER"
- the section called "IEFBR14"
- the section called "JCLBCICS"

- the section called "Utilità del database"
  - the section called "DSNTEP2/DSNTEP4"
  - the section called "DSNUTILB"
  - the section called "INFUTILB/INZUTILB"
  - the section called "JXHDBCLR"
- the section called "Comandi e utilità"
  - the section called "IKJEFT1A/IKJEFT1B/KEQEFT01/IKJEFT01/DSNDBTCH"
  - the section called "QCMDEC"
- the section called "Utilità di ordinamento"
  - the section called "ATTREZZO DA GHIACCIO"
  - the section called "MFSORT"
  - the section called "SORT/SYNCSORT/ICEMAN"
- the section called "Altro/Utilità varie"
  - the section called "AND/CBL_OR/CBL_XOR/CBL_EQ/CBL_IMP/CBLCBL_ _NON"
  - the section called "CEE3ABD"
  - the section called "CREARE"
  - the section called "CEELOCT"
  - the section called "CEERAN (0)"
  - the section called "FORMAGGI"
  - the section called "ILBOABN0"

- Utilità per set di dati
- Utilità del database
- Comandi e utilità
- Utilità di ordinamento
- Altro/Utilità varie

Utilità per set di dati

BLUESAMCOPY/BLUESAMCREATE/BLUESAMDELETE/BLUESAMCLEAR

Questi programmi di utilità forniscono funzioni essenziali di manipolazione dei set di dati sui set di dati Blusam.

Il `bypassBluesamStatus` flag (memorizzato in JobContext) determina se bypassare la normale convalida dello stato del set di dati che impedisce conflitti di accesso simultanei tra processi online e processi batch.

Quando è vero: `bypassBluesamStatus`

- Salta i normali controlli dello stato del set di dati durante le operazioni
- Consente l'accesso a set di dati che normalmente potrebbero essere bloccati

BLUESAMCOPY

Il programma BLUESAMCOPY copia i dati e la struttura dei metadati da un set di dati Bluesam a un altro.

Parameters

Il programma BLUESAMCOPY richiede due parametri:

- `FROM=<source dataset name>`- Set di dati di origine da cui copiare
- `TO=<target dataset name>`- Set di dati di destinazione in cui copiare

Comportamento

- Se il set di dati di destinazione non esiste, lo crea con la struttura di metadati della fonte
- Se il target esiste, lo cancella prima della copia (sovrascrive i dati esistenti)

- L'operazione procede quando i set di dati non sono utilizzati dai processi online o quando è impostato su true `bypassBluesamStatus`
- Conserva le chiavi, la compressione, la lunghezza dei record e altre proprietà dei metadati

Gestione errori

Restituisce il codice 1 se il set di dati di origine non esiste o se uno dei due set di dati è utilizzato dai processi online.

Esempio di utilizzo

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

// copy bluesam dataset
res =
 mpr.withArguments("FROM=MYBLUESAMTABLE","TO=MYBLUESAMTABLE2").runProgram("BLUESAMCOPY")
```

BLUESAM CREATE

Il programma BLUESAMCREATE crea un set di dati Bluesam con i parametri specificati.

Parameters

Il programma BLUESAMCREATE richiede sette parametri:

- `name=<dataset name>`- Set di dati da creare (obbligatorio)
- `compress=<true/false>`- Abilita la compressione (opzionale, impostazione predefinita: false)
- `fixedLength=<true/false>`- Stato dei record a lunghezza fissa (opzionale, impostazione predefinita: false)
- `recordLength=<integer>`- Lunghezza del record in byte (opzionale, impostazione predefinita: -1)
- `primaryKey=<key id>,duplicates=<true/false>,fields=<offset,length,...>`- Specificazione della chiave primaria (opzionale)
- `key=<altkey id>,duplicates=<true/false>,fields=<offset,length,...>`- Chiave alternativa (può essere multipla o 0)

- `clearExisting=true/false`- Cancella il set di dati esistente (impostazione predefinita opzionale: true)

Spiegazioni dei parametri chiave specifici:

- `duplicates`: se i valori duplicati sono consentiti o meno per la chiave data;
- `fields`: posizioni dei campi (in base 1) e lunghezze che definiscono la chiave;

Comportamento

- Se il set di dati non esiste, lo crea con i parametri specificati
- Se il set di dati esiste, lo cancella se `clearExisting=true`
- L'operazione procede quando i set di dati non sono utilizzati dai processi online o quando è impostato su `bypassBluesamStatus` `true`

Condizioni di errore

Restituisce il codice 0 in tutti i casi.

Esempio di utilizzo

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

// create bluesam dataset
mpr.withArguments(
      "name=MYBLUESAMTABLE",
      "compress=FALSE",
      "fixedLength=true",
      "recordLength=54",
      "primaryKey=MYKEY_PK,duplicates=FALSE,fields=0,6")
    .runProgram("BLUESAMCREATE")
```

BLUESAMDELETE

Il programma BLUESAMDELETE elimina i set di dati Bluesam.

Parameters

Il programma BLUESAMDELETE richiede 1 o più parametri:

- `<dataset name>`- set di dati da eliminare (può essere multiplo)

Comportamento

- L'operazione procede quando i set di dati non sono utilizzati dai processi online o quando `bypassBluesamStatus` è impostato su `true`
- Se il set di dati esiste, eliminalo

Condizioni di errore

Restituisce il codice 0 in tutti i casi.

Esempio di utilizzo

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

// delete bluesam datasets
res =
 mpr.withArguments("MYBLUESAMTABLE","MYBLUESAMTABLE2","MYBLUESAMTABLE3").runProgram("BLUESAMDEL
```

BLU AM CLEAR

Il programma BLUESAMCLEAR rimuove tutti i dati dai set di dati esistenti preservandone la struttura e i metadati.

Parameters

Il programma BLUESAMCLEAR richiede 1 o più parametri:

- `<dataset name>`- set di dati da cancellare (può averne più di uno)

Comportamento

- L'operazione procede quando i set di dati non sono utilizzati dai processi online o quando `bypassBluesamStatus` è impostato su `true`
- Se il set di dati esiste, cancellalo

Condizioni di errore

Restituisce il codice 0 in tutti i casi.

Esempio di utilizzo

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

// clear bluesam datasets
res = mpr.withArguments("MYBLUESAMTABLE","MYBLUESAMTABLE2").runProgram("BLUESAMCLEAR")
```

BPXWDYN

Questo programma di utilità simula il servizio z/OS BPXWDYN per l'allocazione e la deallocazione dinamica dei set di dati. Nelle applicazioni moderne, le allocazioni dei file vengono eseguite negli script Groovy tramite builder api. `FileConfigurationUtils` Questo programma consente la gestione dinamica di queste allocazioni senza Groovy, essenziale per l'elaborazione interattiva o condizionale dei file nelle applicazioni modernizzate.

Parameters

Il formato degli argomenti precedenti viene preservato. Prevede un singolo argomento contenente un'intestazione di 2 byte seguita dalla stringa di comando.

formato della stringa di comando per l'allocazione: `ALLOC DD(<name>) DSN('<dsn>') SHR?`

formato della stringa di comando per la deallocazione: `FREE DD(<name>)`

Gestione errori

- Imposta il codice di errore 0 per il successo, 1 per l'errore

- Lancia RuntimeException comandi o parametri non validi

Esempio di utilizzo

Ecco un esempio di utilizzo in Java del programma BPXWDYN, derivante da una modernizzazione COBOL tramite AWS Transform:

Il codice COBOL:

```
   01  WK-AREA.
       03  DS-ALLOC-STRING.
           05  DS-LENGTH          PIC S9(004) COMP VALUE 100.
           05  DS-TEXT            PIC  X(100) VALUE
            "ALLOC DD(INFILE) DSN('A.B.JCLLIB(XYZ470)') SHR".
...
           01  RC-RETURN-CODE-AREA.
       03  RC-RETURN-CODE         PIC S9(008) COMP.
...
     CALL  'BPXWDYN'  USING  DS-ALLOC-STRING RETURNING RC-RETURN-CODE.
```

e la corrispondente modernizzazione di Java:

```
private final Group dsAllocString = new Group(root).named("DS-ALLOC-STRING");
private final Elementary dsLength = new Elementary(dsAllocString,new BinaryType(4, 0,
 "STD", false, false, true),Short.valueOf("100")).named("DS-LENGTH");
private final Elementary dsText = new Elementary(dsAllocString,new
 AlphanumericType(100),"ALLOC DD(INFILE) DSN('A.B.JCLLIB(XYZ470)') SHR").named("DS-
TEXT");
...
private final Group root = new Group(getData()).named("RC-RETURN-CODE-AREA");
private final Elementary rcReturnCode = new Elementary(root,new BinaryType(8, 0, "STD",
 false, false, true)).named("RC-RETURN-CODE");
...
    // Call to utility program  BPXWDYN
ctrl.callSubProgram(
       "BPXWDYN",

 CallBuilder.newInstance().byReference(ctx.getWkArea().getDsAllocStringReference()).getArgument
 ctx);

ctx.getRcReturnCodeArea().setRcReturnCode(NumberUtils.convert(ctx.getProgramReturned()).intValu
```

GDGUTILS

GDGs (Generation Data Group) consente alle applicazioni di lavorare con set di dati con versioni in cui ogni esecuzione crea una nuova generazione mantenendo l'accesso alle generazioni precedenti. Questa utilità crea e gestisce queste generazioni di set di dati. Questa utilità è pensata per essere chiamata negli script groovy.

Parameters

l'ordine dei parametri non ha importanza

- `action=<create|refreshevents>`- Operazione da eseguire (obbligatoria)
- `gdgname=<name>`- Nome della base GDG (obbligatorio)
- `storageProvider=<filesystem|bluesam>`- Backend di archiviazione
- `relativeGeneration=<+integer>`- Numero di generazione relativo (ad esempio, +1 per il prossimo)
- `absoluteGeneration=integer`- Numero di generazione assoluto
- `recordLength=<integer>`- Dimensione del record nel set di dati
- `fixedLength=<true/false>`- Speciifica che i record hanno una lunghezza fissa nel set di dati
- `ownerPath=<File system path>`- Il percorso per memorizzare il set di dati (specifico del file system, obbligatorio in quel caso)
- `compress=<true/false>`- Indica che i dati devono rimanere compressi in memoria se i dati sono stati compressi nel data store (specifico per bluesam) (facoltativo, false è il valore predefinito)
- `catalog`- Speciifica che il set di dati deve essere catalogato (opzionale)
- `warmUp`- Indica che il set di dati deve essere caricato in memoria all'apertura (specifico per bluesam) (opzionale)

RelativeGeneration o AbsoluteGeneration: è necessario impostare una di queste proprietà.

Operazioni

- create: crea una nuova generazione di set di dati GDG in base ai metadati GDG (gestisce sia il filesystem che lo storage Bluesam)
- refreshevents: regola il numero di generazione (Gdg Metadata) senza creare nuovi set di dati (utilizzato quando si riavviano lavori non riusciti che avevano già creato set di dati)

Gestione errori

- Imposta il codice di errore 0 per il successo, 1 per l'errore
- Lancia RuntimeException comandi o parametri non validi

Esempio di utilizzo

Operazione di creazione Gdg: il codice seguente crea una generazione bluesam (43) per il set di dati `IC.PLI.GDGTEST`

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

Map params = new Hashtable()
params.put("jobContext", jobContext)
Object[] args
 =["action=create","gdgname=IC.PLI.GDGTEST","absoluteGeneration=43","storageProvider=bluesam","
mpr.withParameters(params).withArguments(args).runProgram("GDGUTILS")
```

Operazione Gdg refreshevents: solo RelativeGeneration è rilevante per questa operazione. Il codice seguente aggiorna il numero di generazione (+1 rispetto alla generazione attuale) per il set di dati `IC.PLI.GDGTEST`

```
import ...

mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro

Map params = new Hashtable()
params.put("jobContext", jobContext)
Object[] args
 =["action=refreshevents","gdgname=IC.PLI.GDGTEST","relativeGeneration=1","storageProvider=blue
mpr.withParameters(params).withArguments(args).runProgram("GDGUTILS")
```

ICEGENER/SYNCGENR

Questo programma di utilità imita il comportamento dell'utilità di z/OS sistema ICEGENER, copia i set di dati dall'input (set di dati) all'output (set di dati). SYSUT1 SYSUT2 Questa implementazione Java fornisce funzionalità equivalenti che supportano sia il filesystem che l'archiviazione Bluesam.

Parameters

Nessun argomento

Set di dati richiesti

- SYSUT1: set di dati/file di input
- SYSUT2: set di dati/file di output

Gestione della disposizione per il set di dati SYSUT2

- NOVITÀ: crea un nuovo set di dati/file
- OLD/SHR: Use existing dataset/file(deve esistere)
- MOD: Modifica, crea se manca, aggiungi se esiste

Gestione errori

- Imposta il codice di errore 0 se la copia riesce, 1 se fallisce
- Lancia `IllegalStateException` per un utilizzo non valido del set di dati Blusam

Esempio di utilizzo

Ecco un esempio di utilizzo Groovy del programma ICEGENER, derivante da una modernizzazione di JCL tramite AWS Transform:

Il codice JCL:

```
//STEP01    EXEC PGM=ICEGENER
//SYSUT1   DD DSN=POI.INPUT,DISP=SHR
//SYSUT2   DD DSN=POI.OUTPU,
//            DISP=(,CATLG,DELETE),
//            UNIT=3490,
//            DCB=(RECFM=FB,LRECL=100)
/*
```

e la corrispondente modernizzazione di Groovy:

```
mpr
```

```
    .withFileConfigurations(new FileConfigurationUtils()
        .withJobContext(jobContext)
        .bluesam("SYSUT1")
        .dataset("POI.INPUT")
        .disposition("SHR")
        .build()
        .bluesam("SYSUT2")
        .dataset("POI.OUTPU")
        .normalTermination("CATLG")
        .abnormalTermination("DELETE")
        .build()
        .getFileConfigurations())
    .withParameters(params)
    .runProgram("ICEGENER")
```

IDCAMS/KQCAMS

Questo programma di utilità imita il comportamento dei programmi legacy IDCAMS, uno strumento di gestione dei dati mainframe utilizzato per le operazioni sui file VSAM (Virtual Storage Access Method). Elabora i comandi IDCAMS legacy, mantenendo la stessa sintassi degli input SYSIN originali.

Contesto

Il comportamento del programma può essere configurato mediante due parametri definiti in: `application-utility-pgm.yml`

- jclType: identificatore del tipo JCL (vse o mvs). I PRINT/REPRO comandi dell'utilità IDCAMS restituiscono 4 se il file è vuoto per jcl non vse
- forcedCharsetIdcams: Sostituzione opzionale del set di caratteri per l'elaborazione IDCAMS

Parameters

Nessun argomento. Le operazioni vengono passate attraverso il set di dati SYSIN.

Set di dati richiesti

- SYSIN: contiene le istruzioni di comando IDCAMS
- Set di dati di input/output: come indicato nei comandi IDCAMS (a seconda dell'istruzione IDCAMS)

Caratteristiche principali/Comandi supportati

I dettagli sui comandi IDCAMS presenti nella scheda di controllo SYSIN non sono forniti qui, ma devono essere recuperati dalla documentazione esistente delle piattaforme legacy pertinenti.

- DEFINE: crea cluster e set di dati VSAM
- DELETE: rimuove i set di dati (supporta i caratteri jolly)
- REPRO - Copia i dati tra set di dati
- PRINT - Visualizza il contenuto del set di dati
- VERIFY - Convalida l'esistenza e l'integrità del set di dati
- ALTER - Modifica gli attributi del set di dati (ridenominazione)
- ALLOC - Alloca i set di dati in modo dinamico
- SET - Gestisce i codici delle condizioni (LASTCC/MAXCC)
- IF-THEN-ELSE - Esecuzione condizionale di comandi
- ANNULLA - Termina l'esecuzione del lavoro

Gestione errori

- Imposta il codice di errore 0 se l'ultimo comando ha esito positivo, -1 se fallisce
- I comandi SET (LASTCC) possono essere usati per sovrascrivere il codice di errore, ad es. `SET LASTCC = 0`

Esempio di utilizzo

Ecco un esempio di utilizzo Groovy del programma IDCAMS, derivante da una modernizzazione di JCL tramite AWS Transform:

Il codice JCL:

```
//STEP15 EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//ACCTDATA DD DISP=SHR,
//         DSN=AWS.M2.CARDDEMO.ACCTDATA.PS
//ACCTVSAM DD DISP=SHR,
//         DSN=AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS
//SYSIN    DD   *
```

```
  REPRO INFILE(ACCTDATA) OUTFILE(ACCTVSAM)
/*
```

e la corrispondente modernizzazione di Groovy:

```
mpr.withFileConfigurations(new FileConfigurationUtils()
    .withJobContext(jobContext)
    .systemOut("SYSPRINT")
    .output("*")
    .build()
    .bluesam("ACCTDATA")
    .dataset("AWS.M2.CARDDEMO.ACCTDATA.PS")
    .disposition("SHR")
    .build()
    .bluesam("ACCTVSAM")
    .dataset("AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS")
    .disposition("SHR")
    .build()
    .fileSystem("SYSIN")
    .stream("REPRO INFILE(ACCTDATA) OUTFILE(ACCTVSAM)", getEncoding())
    .build()
    .getFileConfigurations())
.withParameters(params)
.runProgram("IDCAMS")
```

IEBGENER/JSDGENER

Questo programma replica l'utilità IEBGENER. Viene utilizzato per copiare e manipolare set di dati sequenziali. Questa implementazione estende le funzionalità di copia di base supportando le istruzioni di controllo IEBGENER per operazioni avanzate di elaborazione dei dati.

Parameters

Nessun argomento. Le operazioni vengono passate attraverso il set di dati SYSIN.

Set di dati richiesti

- SYSIN: contiene istruzioni di controllo (facoltativo, se non definito, il programma IEBGENER è identico a ICEGENER)
- SYSUT1: set di dati/file di input
- SYSUT2: set di dati/file di output

Caratteristiche principali/ Dichiarazioni di controllo IEBGENER supportate

I dettagli sulla dichiarazione di controllo IEBGENER che si trova nelle schede di controllo SYSIN non sono forniti qui, ma devono essere recuperati dalla documentazione esistente delle piattaforme legacy pertinenti.

- GENERATE: definisce la struttura generale del processo di manipolazione dei dati specificando il numero massimo di tipi di record (MAXNAME) e campi (MAXFLDS) da elaborare
- RECORD: definisce il layout e il contenuto effettivi di ogni tipo di record specificando la posizione, la lunghezza e il formato dei singoli campi che verranno copiati dal set di dati di input o generati con valori specifici
- LABEL/MEMBER/EXITnon sono supportati

esempio:

```
GENERATE MAXNAME=3,MAXFLDS=5
RECORD TYPE=1,
       FIELD=(1,1,CH,VALUE='H'),
       FIELD=(2,30,CH,VALUE='EMPLOYEE REPORT 2024         ')
RECORD TYPE=2,
       FIELD=(1,1,CH,VALUE='D'),
       FIELD=(2,10,CH),              /* Name */
       FIELD=(12,8,CH),             /* Birth date */
       FIELD=(20,8,CH,VALUE='ACTIVE')
RECORD TYPE=3,
       FIELD=(1,1,CH,VALUE='F'),
       FIELD=(2,30,CH,VALUE='END OF REPORT             ')
```

Gestione errori

Imposta il codice di errore 0 se la copia riesce, 1 se fallisce.

Esempio di utilizzo

Ecco un esempio di utilizzo Groovy del programma IEBGENER, derivante da una modernizzazione di JCL tramite AWS Transform:

Il codice JCL:

```
//GENDATA  EXEC PGM=IEBGENER
```

```
//SYSUT1   DD DSN=INPUT.EMPLOYEE.DATA,
//            DISP=SHR
//SYSUT2   DD DSN=OUTPUT.EMPLOYEE.FILE,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(TRK,(1,1)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SYSIN    DD *
  GENERATE MAXNAME=2,MAXFLDS=5
  RECORD TYPE=1,
         FIELD=(1,1,CH,VALUE='H'),
         FIELD=(2,30,CH,VALUE='EMPLOYEE SALARY REPORT 2024    ')
  RECORD TYPE=2,
         FIELD=(1,1,CH,VALUE='D'),
         FIELD=(2,10,CH),              /* Name from input */
         FIELD=(12,8,CH),             /* Birth date from input */
         FIELD=(20,8,CH),             /* Dept from input */
         FIELD=(28,8,CH)              /* Salary from input */
/*
```

e la corrispondente modernizzazione di Groovy:

```
mpr
    .withFileConfigurations(new FileConfigurationUtils()
        .withJobContext(jobContext)
        .systemOut("SYSPRINT")
        .output("*")
        .build()
        .bluesam("SYSUT1")
        .dataset("INPUT.EMPLOYEE.DATA")
        .disposition("SHR")
        .build()
        .bluesam("SYSUT2")
        .dataset("OUTPUT.EMPLOYEE.FILE")
        .disposition("NEW")
        .normalTermination("CATLG")
        .abnormalTermination("DELETE")
        .build()
        .fileSystem("SYSIN")
        .stream( """GENERATE MAXNAME=2,MAXFLDS=5
  RECORD TYPE=1,
         FIELD=(1,1,CH,VALUE='H'),
         FIELD=(2,30,CH,VALUE='EMPLOYEE SALARY REPORT 2024    ')
  RECORD TYPE=2,
```

```
        FIELD=(1,1,CH,VALUE='D'),
        FIELD=(2,10,CH),              /* Name from input */
        FIELD=(12,8,CH),             /* Birth date from input */
        FIELD=(20,8,CH),             /* Dept from input */
        FIELD=(28,8,CH)              /* Salary from input */""", getEncoding())
       .build()
       .getFileConfigurations())
    .withParameters(params)
    .runProgram("IEBGENER")
```

IEFBR14

IEFBR14 è un programma «non fare nulla» che restituisce semplicemente un codice di ritorno pari a 0 (zero). Il suo utilizzo principale è l'allocazione, l'eliminazione o la manutenzione del catalogo dei set di dati tramite istruzioni DD, senza eseguire alcuna elaborazione effettiva dei dati.

Parameters

Nessun argomento

Gestione errori

Imposta sempre il codice di errore 0.

Esempio di utilizzo

Ecco un esempio di utilizzo del IEFBR14 programma Groovy, derivante da una modernizzazione di JCL tramite AWS Transform:

Il codice JCL per creare un nuovo set di dati sequenziale:

```
//STEP1    EXEC PGM=IEFBR14
//NEWSEQ   DD DSN=USER.NEW.SEQ.DATA,
//            DISP=(NEW,CATLG,DELETE),
//            SPACE=(TRK,(10,5)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
```

e la corrispondente modernizzazione di Groovy:

```
mpr
    .withFileConfigurations(new FileConfigurationUtils()
        .withJobContext(jobContext)
        .bluesam("NEWSEQ")
```

```
        .dataset("USER.NEW.SEQ.DATA")
        .disposition("NEW")
        .normalTermination("CATLG")
        .abnormalTermination("DELETE")
        .build()
        .getFileConfigurations())
    .withParameters(params)
    .runProgram("IEFBR14")
```

Il codice JCL per eliminare un set di dati esistente:

```
//STEP3    EXEC PGM=IEFBR14
//DELDD    DD DSN=USER.OLD.DATASET,
//            DISP=(OLD,DELETE,DELETE)
```

e la corrispondente modernizzazione di Groovy:

```
mpr
    .withFileConfigurations(new FileConfigurationUtils()
        .withJobContext(jobContext)
        .bluesam("DELDD")
        .dataset("USER.OLD.DATASET")
        .disposition("OLD")
        .normalTermination("DELETE")
        .abnormalTermination("DELETE")
        .build()
        .getFileConfigurations())
    .withParameters(params)
    .runProgram("IEFBR14")
```

JCLBCICS

Questo programma gestisce lo stato dei set di dati, i enables/disables set di dati sono basati sulla configurazione, supportando sia i singoli file che i pattern wildcard: modifica il campo STATUS nella tabella JICS file_table.

Parameters

Nessun argomento. Le operazioni vengono passate attraverso un `DatasetsConfiguration` oggetto, ad es.

```
mpr.withDatasetsConfiguration(new DatasetsConfiguration().close(<Dataset Name>)
```

Contesto

Il comportamento del programma può essere configurato mediante due parametri:

In `application-utility-pgm.yml`:

- `jclbcics.ddname.size`(l'impostazione predefinita è 8): configura globalmente la dimensione del nome del set di dati; se questo valore è impostato e la lunghezza del nome del set di dati è inferiore a questo valore, il nome del set di dati verrà troncato.

In un singolo passaggio del file Groovy che chiama il programma

- `JCLBCICS_OVERRIDDEN_SIZE`: sovrascrive la dimensione globale del nome del set di dati:

```
TreeMap stepMapTransfo = [:]
Map stepParams = ["MapTransfo":stepMapTransfo]
stepParams["MapTransfo"]["JCLBCICS_OVERRIDDEN_SIZE"] = '6'
```

Se la dimensione corretta del nome dd (dopo la troncatura) è inferiore a 8, il nome dd viene considerato come una wild card e la funzione funziona per tutti i set di dati che iniziano con questo nome dd.

Caratteristiche chiave

Le operazioni supportate sono:

- APERTO: imposta i set di dati allo stato ABILITATO
- CHIUDI: imposta i set di dati allo stato DISABILITATO

Queste operazioni sono dichiarate tramite la classe `DatasetsConfiguration` builder:

```
new DatasetsConfiguration().close(<DD name>).open(<DD name>)
```

Nome DD: nome del set di dati, il carattere jolly * è accettato se la dimensione del nome dd è inferiore alla dimensione massima del nome dd (8).

Gestione errori

Imposta il codice di errore 0

Esempio di utilizzo

Ecco un esempio di utilizzo di Groovy del programma JCLBCICS:

disabilita il set di dati, abilita tutti i set di dati che iniziano con AXUF0LJ3P, abilita tutti i set di dati che iniziano con DX

```
import com.netfective.bluage.gapwalk.rt.call.MainProgramRunner
import com.netfective.bluage.gapwalk.rt.call.ProgramExecutionResult
import com.netfective.bluage.gapwalk.rt.io.support.DatasetsConfiguration

MainProgramRunner mpr =
 applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController",MainPro
def TreeMap stepMapTransfo = [:]
def Map stepParams = ['MapTransfo':stepMapTransfo]
stepParams['MapTransfo']['JCLBCICS_OVERRIDDEN_SIZE'] = '7'

ProgramExecutionResult res = mpr
.withDatasetsConfiguration(new
 DatasetsConfiguration().close("UF0LJ3P").open("AX*").open("DX"))
.withParameters(stepParams)
.runProgram("JCLBCICS")
```

Utilità del database

Questa sezione riguarda le utilità relative ai database. Sulle piattaforme legacy, questi programmi di solito operano su DB2 database, mentre vengono scelti database alternativi (AWS Aurora è una scelta popolare) su ambienti moderni.

DSNTEP2/DSNTEP4

DSNTEP2/Scopo DSNTEP4

L'utilità DNSTEP esegue query SQL dai set di dati di input e scrive i risultati nei set di dati di output. Riscrive automaticamente le query SQL legacy per database di destinazione modernizzati (oltre DB2) e supporta diversi tipi di set di dati: stream, SYSIN set di dati Blusam, file flat, generazioni GDG e concatenazioni di set di dati.

DSNTEP2/Firma DSNTEP4

L'utilità non accetta argomenti e utilizza due set di dati:

- `SYSIN`: set di dati di input contenente istruzioni SQL
- `SYSPRINT`: set di dati di output per i risultati delle query (solo `SELECT` query)

DSNTEP2/parametri di DSNTEP4 configurazione correlati

Il comportamento dell'utilità è influenzato dai seguenti parametri di configurazione:

- `unload.sqlCodePointShift`
- `unload.noPad`
- `unload.nbi.whenNull`
- `unload.nbi.whenNotNull`
- `unload.useDatabaseConfiguration`
- `unload.format.date`
- `unload.format.time`
- `unload.format.timestamp`
- `hasGraphic`
- `forcedDate`
- `frozenDate`

the section called "Proprietà disponibili per applicazioni Web opzionali"Per ulteriori informazioni sulla configurazione di questi parametri, consulta la sezione.

DSNTEP2/DSNTEP4 Controlli/Gestione degli errori

- Se, per qualsiasi motivo, si verifica un'eccezione durante l'esecuzione della query, verrà registrato un messaggio di errore e `StopRunUnitException` verrà generato un messaggio che porterà all'arresto dell'unità di esecuzione corrente.
- Se si `SYSIN` tratta di una concatenazione di vari set di dati e uno qualsiasi dei set di dati non è supportato, verrà generato un. `RuntimeException` Attualmente, solo i file flat e i tipi di generazioni GDG sono supportati come parti di un file concatenato, se utilizzati come input con /4. DSNTEP2

DSNTEP2/Esempi di utilizzo DSNTEP4

Ecco un esempio di utilizzo di JCL di: DNSTEP4

```
//*******************************************************************
//* RETRIEVE DATA FROM TABLE AP_JBI7_INVOICE                        *
//*******************************************************************
//*
//DSNTEP03 EXEC PGM=DSNTEP4,DYNAMNBR=20
//SYSPRINT DD DSN=output(out012.txt),
//            DISP=SHR,DCB=(RECFM=FB,LRECL=1152)
//SYSIN    DD *
  SELECT * FROM BUR000.AP_JBI7_INVOICE WITH UR;
```

e lo snippet di script groovy modernizzato corrispondente:

```
def stepDSNTEP03(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("DSNTEP03", "DSNTEP4", programResults, {
                    mpr
                        .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .fileSystem("SYSPRINT")
                        .path("output(out012.txt)").recordSize(1152)
                        .disposition("SHR")
                        .build()
                        .fileSystem("SYSIN")
                        .stream(
"""  SELECT * FROM BUR000.AP_JBI7_INVOICE WITH UR;
  """, getEncoding())
                        .build()
                        .getFileConfigurations(fcmap))
                        .withParameters(params)
                    .runProgram("DSNTEP4")
                })
        }
    }
}
```

Si noti che la query SQL viene fornita «così com'è» all'utilità /4, senza alcuna modifica. DSNTEP2

DSNUTILB

Scopo DSNUTILB

Utilità di database per il caricamento, la copia e la gestione dei dati. Di solito funziona su DB2 database legacy; la versione modernizzata supporta AWS Aurora e altri database di destinazione.

Firma DSNUTILB

Per sua natura, DSNUTILB è destinato piuttosto a essere chiamato da script di lavoro modernizzati.

Non accetta argomenti; legge i comandi dal set di dati. `SYSIN`

I comandi supportati sono:

- `TEMPLATE`(allocazione dinamica dei set di dati)
- `LISTDEF`(raggruppa gli oggetti del database in elenchi, utilizzabili con altri comandi)
- `COPY`(creare copie degli oggetti del database)
- `LOAD`(carica i record nelle tabelle)
- `DISCARD`(eliminare i record dalle tabelle)

Per ulteriori dettagli sui comandi, si prega di consultare la relativa documentazione preesistente.

Parametri di configurazione relativi a DSNUTILB

Il comportamento dell'utilità è influenzato dai seguenti parametri di configurazione:

- `unload.useDatabaseConfiguration`
- `load.format.localDate`
- `load.format.dbDate`
- `load.format.localTime`
- `load.format.dbTime`
- `load.sqlCodePointShift`
- `convertGraphicDataToFullWidth`

the section called “Proprietà disponibili per applicazioni Web opzionali”Per ulteriori informazioni sulla configurazione di questi parametri, consulta la sezione.

Controlli DSNUTILB /Gestione degli errori

- Se il `SYSIN` set di dati non contiene alcun comando utilizzabile, verrà generato un`RuntimeException`.
- Se si verifica un'eccezione durante le operazioni del database, verrà registrato un messaggio di errore, il codice restituito verrà impostato su 8 e `StopRunUnitException` verrà generato un (arrestando l'unità di esecuzione corrente).

DSNUTILB Esempi di utilizzo

Ecco un esempio di utilizzo di DSNUTILB in uno script JCL:

```
//*******************************************************************
//* LOAD DATA IN TABLE AP_JBI7_INVOICE.                             *
//*******************************************************************
//DSN01  EXEC PGM=DSNUTILB,DYNAMNBR=20
//SYSREC DD  DSN=input(input021.data),
//           DISP=SHR
//           DCB=(RECFM=FB,LRECL=76)
//SYSIN  DD  DSN=input(dsn01.card),
//           DISP=SHR
```

con i comandi card content (dsn01.card) -- usati per caricare i dati nel database, da un file flat esportato dalla piattaforma precedente --:

```
  LOAD         DATA
               INDDN         SYSREC
               RESUME        NO
               LOG           YES
               NOCOPYPEND
               SORTDEVT      SYSDA
               SORTNUM       12
               SORTKEYS      100000
               DISCARDS      0
               INTO TABLE    BUR000.AP_JB17_INVOICE
               WHEN (76:76) = 'L'
( IDENTIFIER         POSITION(1:1)        SMALLINT
 ,CUST_ID            POSITION(10)         VARCHAR
                                          NULLIF(39) = '?'
 ,CUST_KD            POSITION(40:43)      CHAR
 ,INVC_AMNT          POSITION(44:49)      NUMERIC
```

```
  ,INVC_DAT          POSITION(50:75)      TIMESTAMP EXTERNAL(26)
 )
```

e lo snippet di script groovy modernizzato corrispondente:

```
// STEP DSN01 - PGM - DSNUTILB*****************************************************
def stepDSN01(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("DSN01", "DSNUTILB", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .fileSystem("SYSREC")
                        .path("input(input021.data)").recordSize(76)
                        .disposition("SHR")
                        .build()
                        .fileSystem("SYSIN")
                        .path("input(dsn01.card)")
                        .disposition("SHR")
                        .build()
                        .getFileConfigurations(fcmap))
                    .withParameters(params)
                    .runProgram("DSNUTILB")
                    })
            }
    }
}
```

INFUTILB/INZUTILB

Scopo INFUTILB

INFUTILB/INZUTILB sono programmi di utilità utilizzati per estrarre dati dai database -- unload -- (in ambiente legacy) e convertirli DB2 in vari formati di output.

Le query SQL legacy vengono adattate automaticamente on-the-fly per soddisfare i requisiti dei database di destinazione moderni (motori supportati: PostgreSQL, Oracle e). DB2

Firma INFUTILB /INZUTILB

È possibile utilizzare i seguenti alias del programma (che corrispondono al nome delle utilità di ordinamento precedenti corrispondenti):

- INFUTILB
- INZUTILB

L'utilità non accetta alcun argomento ma legge i comandi da eseguire dal SYSIN set di dati (una «scheda di controllo»). I record estratti dal database vengono scritti nel set di dati e il SYSREC set di dati opzionale SYSPUNCH viene utilizzato per memorizzare la scheda di controllo che può essere utilizzata per ricaricare i dati (utilizzando altre utilità, ad esempio). the section called "DSNUTILB"

Per loro natura, INFUTILB/INZUTILB sono destinati principalmente a essere chiamati da script di lavoro modernizzati.

Per ottenere dettagli sui comandi supportati, consulta la documentazione precedente appropriata. INFUTILB/INZUTILB Utilizzano set di dati «schede di controllo» legacy «così come sono».

Parametri di configurazione relativi a INFUTILB /INZUTILB

Il comportamento dell'utilità è influenzato dai seguenti parametri di configurazione:

- unload.sqlCodePointShift
- unload.noPad
- unload.nbi.whenNull
- unload.nbi.whenNotNull
- unload.useDatabaseConfiguration
- unload.format.date
- unload.format.time
- unload.format.timestamp
- unload.columnFiller
- unload.varCharIsNull
- unload.DFSIGDCB
- hasGraphic
- forcedDate
- frozenDate

the section called "Proprietà disponibili per applicazioni Web opzionali"Per ulteriori informazioni sulla configurazione di questi parametri, consulta la sezione.

INFUTILB//INZUTILB Checks//Gestione degli errori

- Se il database di destinazione non fa parte dei motori di database supportati (PostgreSQL, Oracle DB2 e), il codice di ritorno del programma verrà impostato su 8 e verrà generato un. `UnsupportedOperationException`
- Se il programma non riesce a eliminare i file temporanei, il codice restituito verrà impostato su 4, verrà registrato un messaggio di errore, ma l'esecuzione del programma non verrà interrotta.
- In tutti i casi seguenti, il codice di ritorno del programma verrà impostato su 4 o 8 e `AbendException` verrà generato un messaggio (interrompendo l'esecuzione del programma):
  - Se il `SYSREC` set di dati non è uno dei tipi supportati (GDG o set di dati basato su file system); restituisce il codice 4;
  - Se il `SYSPUNCH` set di dati non è uno dei tipi supportati (GDG o set di dati basato su file system o); codice di ritorno 4; `DUMMY`
  - Se il programma non è in grado di recuperare la dimensione del record del `SYSREC` set di dati (non impostata o non definita nel catalogo dei set di dati); codice di ritorno 8;
  - Se il programma non è in grado di recuperare la dimensione del record del `SYSPUNCH` set di dati (non impostata o non definita nel catalogo dei set di dati); codice di ritorno 8;
  - Se la query utilizzata per creare il contenuto del `SYSREC` set di dati non è valida (la query difettosa verrà registrata); codice di ritorno 4;
  - Se si verifica un'eccezione durante il recupero dei dati dal database; codice di ritorno 8;
  - Se il `OUTDDN` comando non è presente nel `SYSIN` set di dati, per un servizio di scarico; restituisci il codice 8;
  - Se non è stato possibile trovare alcun comando valido nel `SYSIN` set di dati; restituisce il codice 8;

INFUTILB//INZUTILB Esempi di utilizzo

Ecco un esempio di frammento di script jcl precedente:

```
//*******************************************************************
//* UNLOAD DATA FROM TABLE AP_JBI7_INVOICE.                         *
//*******************************************************************
//INF1   EXEC PGM=INFUTILB
//SYSREC DD  DSN=output(out032.data),
//           DISP=SHR
//           DCB=(RECFM=FB,LRECL=90)
```

```
//SYSIN  DD  DSN=input(inf12.card),
//           DISP=SHR
```

che utilizza la seguente scheda di comandi (inf12.card) per scaricare alcuni dati dal database (qui, i record vengono selezionati in base alla loro data):

```
  UNLOAD
      SELECT * FROM BUR000.AP_JB17_INVOICE
      WHERE INVC_DAT >=
      CONCAT(STRIP(CHAR(YEAR(CURRENT DATE - 100 YEAR))),'-01-01-00.00.00.000000')
      AND INVC_DAT >=
      CONCAT('2025-01-01-',CONCAT((CURRENT TIME),'.000000'))
      AND INVC_DAT >= (CURRENT TIMESTAMP - 100 YEAR)
      ORDER BY identifier ASC
        OUTDDN (SYSREC)
        FORMAT DSNTIAUL
```

e lo snippet di script groovy corrispondente, risultato della modernizzazione automatizzata di jcl:

```
// STEP INF1 - PGM - INFUTILB*****************************************************
def stepINF1(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("INF1", "INFUTILB", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .fileSystem("SYSREC")
                        .path("output(out032.data)").recordSize(90)
                        .disposition("SHR")
                        .build()
                        .fileSystem("SYSIN")
                        .path("input(inf12.card)")
                        .disposition("SHR")
                        .build()
                        .getFileConfigurations(fcmap))
                    .withParameters(params)
                    .runProgram("INFUTILB")
                })
        }
    }
}
```

JXHDBCLR

Scopo JXHDBCLR

JXHDBCLR è un programma di utilità per la cancellazione di database, disponibile su GS21 piattaforme, che tronca le tabelle in base alle istruzioni specifiche fornite e presenti in una scheda di comando.

Firma JXHDBCLR

Non accetta alcun argomento ma legge le istruzioni dal `SYSIN` set di dati (la scheda dei comandi).

Per sua natura, è destinato principalmente a essere chiamato da script di lavoro modernizzati.

Parametri di configurazione relativi a JXHDBCLR

Il comportamento dell'utilità è influenzato dai seguenti parametri di configurazione:

- `unload.sqlCodePointShift`
- `unload.noPad`
- `unload.nbi.whenNull`
- `unload.nbi.whenNotNull`
- `unload.useDatabaseConfiguration`
- `unload.format.date`
- `unload.format.time`
- `unload.format.timestamp`
- `hasGraphic`
- `forcedDate`
- `frozenDate`

the section called “Proprietà disponibili per applicazioni Web opzionali”Per ulteriori informazioni sulla configurazione di questi parametri, consulta la sezione.

Controlli JXHDBCLR /Gestione degli errori

Se non viene trovata alcuna tabella da troncare, verrà registrato un messaggio di avviso, ma l'esecuzione del programma non verrà interrotta.

Se si verificano alcuni errori durante il troncamento delle tabelle, il codice di ritorno del programma verrà impostato su 4, i messaggi di errore verranno registrati, ma l'esecuzione del programma non verrà interrotta.

Per una qualsiasi delle seguenti condizioni, il codice di ritorno del programma verrà impostato su 8 e `AbendException` verrà generato un (interrompendo l'esecuzione del programma):

- Se il contenuto della scheda comandi è vuoto;
- se si verifica un'eccezione durante l'analisi o l'elaborazione dei comandi;

Esempi di utilizzo di JXHDBCLR

Un esempio di utilizzo di JXHDBCLR con una scheda di comandi incorporata:

```
//*****************************************************************
//**  Step 1 - JXHDBCLR UTILITY - DBCLEAR TYPE=2/3
//*****************************************************************
//STEP01  EXEC PGM=JXHDBCLR,REGION=256K,PARM='LINECNT=0'
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DBCLEAR    SCHEMA=MUSICSCH,TYPE=3,CHECK=YES
    DEFINE     RANGE=(SINGERDEST)
END
```

e lo snippet di script groovy modernizzato corrispondente:

```
// STEP STEP01 - PGM - JXHDBCLR*************************************************
def stepSTEP01(Object shell, Map params, Map programResults){
 shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("STEP01", "JXHDBCLR", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .systemOut("SYSPRINT")
                        .output("*")
                        .build()
                        .fileSystem("SYSIN")
                        .stream(
"""    DBCLEAR    SCHEMA=MUSICSCH,TYPE=3,CHECK=YES
    DEFINE     RANGE=(SINGER)
```

```
END""", getEncoding())
                        .build()
                        .getFileConfigurations())
                    .withArguments(getParm("LINECNT=0"))
                    .withParameters(params)
                    .runProgram("JXHDBCLR")
                })
        }
    }
}
```

La scheda incorporata di Legacy viene mantenuta «così com'è», utilizzando uno stream.

Comandi e utilità

Questa sezione è relativa ai programmi di utilità il cui ruolo è gestire i comandi utente, forniti tramite schede di controllo.

IKJEFT1A/IKJEFT1B/KEQEFT01/IKJEFT01/DSNDBTCH

IKJEFT1Uno scopo

IKJEFT1A e i suoi alias eseguono i comandi TSO (TIME S Sharing Option) in processi batch senza richiedere una sessione TSO interattiva, collegando ambienti batch non interattivi e ambienti TSO interattivi.

Imita il comportamento precedente di IKJEFT1 A con differenze specifiche dell'ambiente: ad esempio, i comandi DB2 TERM (inate) vengono ignorati negli ambienti moderni (registrati come informativi).

> **Note**
>
> Sintassi dei comandi non dettagliata qui: A utilizza i set di dati dei comandi legacy invariati. IKJEFT1 Consulta la documentazione della piattaforma precedente per i dettagli dei comandi TSO.

IKJEFT1Una firma

Utilizza il `SYSTSIN` set di dati per l'input dei comandi (definito tramite la `DD` direttiva JCL). La versione modernizzata utilizza lo stesso set di dati invariato. Richiamato solo negli script Groovy jobs modernizzati.

IKJEFT1Un parametro di configurazione correlato

Il comportamento dell'utilità è influenzato dal seguente parametro di configurazione:

- `systin.encoding`

the section called "Proprietà disponibili per applicazioni Web opzionali"Per ulteriori informazioni sulla configurazione di questo parametro, consulta la sezione.

IKJEFT1Una gestione Checks/Error

Quando un comando non supportato è presente nel set di dati SYSTSIN, viene generato un`RuntimeException`.

IKJEFT1A) Esempi di utilizzo

Lo script JCL legacy (che utilizza un contenuto in linea del `SYSTSIN` set di dati, utilizzando la direttiva) `DD *`

```
//*******************************************************************
//* READ THE TEMPORARY INPUT FILE                                   *
//*******************************************************************
// IF IDCM00032.RC = 1 THEN
//067FILEKEY  EXEC PGM=IKJEFT01,DYNAMNBR=20
//SYSTSIN   DD *
 DSN SYSTEM(DB2P)
 RUN  PROGRAM(067-fileKey) PLAN(FILEKEYPLAN)    PARM('RT')
 END
/*
//TEMPVSAM  DD DSN=IDXVIDEO.TEMPVSAM,DISP=SHR
//FKOUT     DD DSN=output(out067.txt),DISP=(NEW,CATLG)
// ENDIF
//*******************************************************************
```

e lo script modernizzato corrispondente (in groovy) - il contenuto in linea del `SYSTSIN` set di dati è rappresentato da un «flusso» -

```
// STEP 067FILEKEY - PGM - IKJEFT01***********************************************
def step067FILEKEY(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("067FILEKEY", "IKJEFT01", programResults, {
```

```
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .fileSystem("SYSTSIN")
                        .stream(
""" DSN SYSTEM(DB2P)
 RUN  PROGRAM(067-fileKey) PLAN(FILEKEYPLAN)    PARM('RT')
 END
""", getEncoding())
                        .build()
                        .bluesam("TEMPVSAM")
                        .dataset("IDXVIDEO.TEMPVSAM")
                        .disposition("SHR")
                        .build()
                        .fileSystem("FKOUT")
                        .path("output(out067.txt)")
                        .disposition("NEW")
                        .normalTermination("CATLG")
                        .build()
                        .getFileConfigurations())
                    .withParameters(params)
                    .runProgram("IKJEFT01")
                })
        }
    }
}
```

QCMDEC

Scopo QCMDEXC

Questo programma di utilità emula il comportamento dell'utilità di sistema AS/400 QCMDEXC, utilizzata per eseguire i comandi di sistema in modo dinamico in fase di esecuzione.

Funzionalità:

- Analizza ed esegue i comandi in modo dinamico.
- Supporta i formati di parametri denominati e posizionali.

Firma QCMDEXC

Accetta 1-2 parametri:

- Primo: dati alfanumerici contenenti comandi da eseguire
- Secondo (opzionale): lunghezza in byte da leggere dal primo parametro (il valore predefinito è la lunghezza completa)

Gestione QCMDEXC Checks/Error

Nei seguenti casi, `RuntimeException` verrà generato un:

- Quando il numero di argomenti forniti non è uno o due;
- Se l'argomento dei comandi è vuoto o se il comando da eseguire non è riconosciuto;
- Se l'esecuzione del comando fallisce per qualsiasi motivo; inoltre, il codice restituito dal programma verrà impostato su 1.

QCMDEXC Esempi di utilizzo

Ecco un esempio di utilizzo di un programma COBOL legacy: l'elemento di dati `CL-COMMANDX` è dotato di comandi prima di essere utilizzato come argomento per la chiamata al programma QCMDEXC:

```
       INITIALIZE CL-COMMANDX.

       STRING
           "CPYF FROMFILE(PLAYERS) TOFILE(" DELIMITED BY SIZE,
           "NEWFILE"                        DELIMITED BY SIZE,
           ") MBROPT(*REPLACE) FMTOPT(*NOCHK) CRTFILE(*YES)"
                                            DELIMITED BY SIZE
           INTO CL-COMMANDX
       END-STRING.

       CALL "QCMDEXC"                  USING CL-COMMANDX
                                             CLENGTHX.
```

Una volta modernizzato in codice java, questo diventa:

```
    DataUtils.initialize(ctx.getClCommandx().getClCommandxReference());

StringConcatenationBuilder.newInstance(ctx.getClCommandx().getClCommandxReference())
       .addDelimitedBySize("CPYF FROMFILE(PLAYERS) TOFILE(")
       .addDelimitedBySize("NEWFILE")
```

```
        .addDelimitedBySize(") MBROPT(*REPLACE) FMTOPT(*NOCHK) CRTFILE(*YES)")
        .end();
    ctrl.callSubProgram("QCMDEXC", CallBuilder.newInstance()
        .byReference(ctx.getClCommandx().getClCommandxReference())
        .byReference(ctx.getClengthx().getClengthxReference())
        .getArguments(), ctx);
```

Tieni presente che il contenuto dei comandi precedenti viene utilizzato «così com'è» nel codice modernizzato, senza alcuna modifica.

Utilità di ordinamento

Le utilità di ordinamento si basano su set di dati il cui riferimento ha un significato speciale per l'utilità. Quindi, tutte le utilità di ordinamento sul lato modernizzato condividono lo stesso schema di invocazione negli script groovy:

```
mpr.withFileConfigurations(<FILE CONFIGURATIONS DETAILS>...)
   .withParameters(params)
   .runProgram("<SORT UTILITY ALIAS>")
```

dove `mpr` è un'`MainProgramRunner`istanza (vedi la sezione Scripts calling programs in questa pagina). Gli alias delle utilità di ordinamento sono riportati nelle sezioni seguenti.

Le voci `DD` (definizioni dei set di dati) dello script JCL vengono modernizzate come voci nei dettagli della configurazione del file che fungono da argomento per il metodo. `mpr.withFileConfigurations()` Vedi gli esempi seguenti per illustrazioni di questo concetto.

SORT/SYNCSORT/ICEMAN

Scopo

Questo programma emula varie utilità SORT per mainframe, utilizzate per sort/merge/copy i dati dei set di dati, in base ai criteri forniti. È possibile utilizzare i seguenti alias di programma (che corrispondono al nome dell'utilità di ordinamento esistente corrispondente):

- `SORT`
- `SYNCSORT`
- `ICEMAN`

I dettagli sulle SORT/MERGE direttive presenti nelle schede di controllo e sulle funzionalità legacy dell'utilità di ordinamento non sono forniti qui, ma devono essere recuperati dalla documentazione esistente sulle piattaforme legacy pertinenti.

Firma

Il programma non accetta alcuna argomentazione ma si basa invece su riferimenti a set di dati specifici:

- Il `SYSIN` set di dati (noto anche come «scheda di controllo») contiene le istruzioni di controllo sort/merge
- Il `SYMNAMES` set di dati opzionale contiene direttive di sostituzione delle variabili nel contenuto SYSIN
- L'opzionale `SORTXDUP` o il `SORTXSUM` set di dati possono essere utilizzati per archiviare record duplicati
- I set di dati sono `SORTDBIN` preceduti `SORTIN` o contengono i record da elaborare (input)
- Il `SORTOUT` set di dati contiene i risultati del programma (output)
- Le `SORTWK` definizioni per i set di dati SORT WORK presenti in alcuni script di job legacy vengono ignorate (e non sono rappresentate nella chiamata modernizzata); i set di dati di sort work verranno sempre allocati dinamicamente nell'ambiente moderno
- I due set di dati il cui DD inizia con `SORTJN` (prefisso) contengono record che riguarderanno le direttive join keys (utilizzate per unire i set di dati durante il processo di ordinamento)

Ad esempio, considerando le seguenti direttive sulle chiavi di unione:

```
JOINKEYS FILE=F1,FIELDS=(13,5,A)
JOINKEYS FILE=F2,FIELDS=(24,5,A)
```

Qui, la chiave di unione ha una lunghezza 5 e inizia da:

- posizione 13 per i record nel set di dati `SORTJNF1` (concatenazione di `SORTJN` prefisso e file) `F1`
- posizione 24 per i record nel set di dati (concatenazione di prefisso e `SORTJNF2` file) `SORTJN` `F2`

Controlli/Gestione degli errori

- Se il file di input (`SORTIN`) ha una `SHR` disposizione ma non può essere trovato, viene registrato un messaggio di errore, il codice di ritorno del programma viene impostato su 1 e l'esecuzione

del programma viene interrotta (non verrà eseguito alcun ordinamento, non verrà prodotto alcun output)

Nei seguenti casi, verrà lanciato un messaggio `RuntimeException` contenente un messaggio dedicato:

- Se l'invocazione del programma richiede la connessione a un database (quando si utilizza il `SORTDBIN` set di dati, no`SORTIN`), è possibile trovare una fonte di dati non valida
- Se il file di output (`SORTOUT`) non è definito correttamente
- Se un comando trovato nella scheda di controllo non può essere compreso o non è supportato
- In caso contrario, vengono forniti due file di input per `SORT JOINKEYS` ogni caso

Esempio di utilizzo

Esempio MERGE

Ecco un esempio di `ICEMAN` richiamo da uno snippet di job script:

La scheda di controllo è in linea e contiene comandi per unire i campi dai file di input (vedi la voce) `SYSIN`

```
//*
//PASOSO03 EXEC PGM=ICEMAN,REGION=0M
//SORTIN01 DD DSN=input(input809a.data),DISP=SHR,LRECL=10
//SORTIN02 DD DSN=input(input809b.data),DISP=SHR,LRECL=10
//SORTOUT  DD DSN=output(out809.txt),DISP=(,PASS),LRECL=10
//SORTWK01 DD SPACE=(281,(156300,156300),RLSE),AVGREC=U
//SORTWK02 DD SPACE=(281,(156300,156300),RLSE),AVGREC=U
//SYSIN    DD *
   MERGE  FIELDS=(1,6,PD,A,7,2,CH,A)
   END
/*
```

E il corrispondente frammento di script groovy modernizzato: tenete presente che, come già detto, le `SORTWK` voci non vengono prese in considerazione durante il processo di modernizzazione e che la scheda di controllo in linea corrisponde esattamente al contenuto della scheda di controllo precedente.

```
// STEP PASOSO03 - PGM - ICEMAN**************************************************
```

```
def stepPASOSO03(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("PASOSO03", "ICEMAN", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .fileSystem("SORTIN01")
                        .path("input(input809a.data)").recordSize(10)
                        .disposition("SHR")
                        .build()
                        .fileSystem("SORTIN02")
                        .path("input(input809b.data)").recordSize(10)
                        .disposition("SHR")
                        .build()
                        .fileSystem("SORTOUT")
                        .path("output(out809.txt)").recordSize(10)
                        .normalTermination("PASS")
                        .build()
                        .fileSystem("SYSIN")
                        .stream(
"""   MERGE  FIELDS=(1,6,PD,A,7,2,CH,A)
   END                                                                """,
 getEncoding())
                        .build()
                        .getFileConfigurations())
                        .withParameters(params)
                    .runProgram("ICEMAN")
                })
        }
    }
}
```

Esempio di SORT semplice

Un semplice passaggio SORT preesistente (job script snippet) con scheda di controllo integrata, tratto dall'applicazione di esempio carddemo:

```
//*****************************************************************
//* CREATE COPY OF TRANSACT FILE WITH CARD NUMBER AND TRAN ID AS KEY
//*****************************************************************
//STEP010  EXEC PGM=SORT
//SORTIN   DD  DISP=SHR,DSN=AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS
```

```
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SORTOUT  DD  DSN=AWS.M2.CARDDEMO.TRXFL.SEQ,
//             DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//             DCB=(LRECL=350,BLKSIZE=3500,RECFM=FB),
//             SPACE=(CYL,(1,1),RLSE)
//SYSIN    DD *
  SORT FIELDS=(263,16,CH,A,1,16,CH,A)
  OUTREC FIELDS=(1:263,16,17:1,262,279:279,50)
/*
```

e lo snippet di script groovy modernizzato corrispondente:

```
// STEP STEP010 - PGM - SORT*************************************************
def stepSTEP010(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("STEP010", "SORT", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .bluesam("SORTIN")
                        .dataset("AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS")
                        .disposition("SHR")
                        .build()
                        .systemOut("SYSPRINT")
                        .output("*")
                        .build()
                        .systemOut("SYSOUT")
                        .output("*")
                        .build()
                        .fileSystem("SORTOUT")
                        .path("AWS.M2.CARDDEMO.TRXFL.SEQ").recordSize(350)
                        .disposition("NEW")
                        .normalTermination("CATLG")
                        .abnormalTermination("DELETE")
                        .build()
                        .fileSystem("SYSIN")
                        .stream(
"""  SORT FIELDS=(263,16,CH,A,1,16,CH,A)
  OUTREC FIELDS=(1:263,16,17:1,262,279:279,50)""", getEncoding())
                        .build()
                        .getFileConfigurations())
```

```
                    .withParameters(params)
                    .runProgram("SORT")
                })
        }
    }
}
```

Si noti che la scheda di controllo integrata viene utilizzata «così com'è», senza alcuna modifica rispetto al contenuto della scheda di controllo precedente.

ATTREZZO DA GHIACCIO

Scopo

L'utilità ICETOOL viene utilizzata per eseguire più operazioni sui set di dati in un'unica fase di lavoro (manipolazione, ordinamento e analisi dei dati).

Sono supportati i seguenti operatori principali:

- `COPY`- Copia i dati dai file di input a quelli di output
- `SORT`- Ordina i dati utilizzando schede/criteri di ordinamento specificati
- `SELECT`- Filtra e seleziona record specifici in base alle condizioni
- `SPLICE`- Merges/joins dati provenienti da più fonti
- `COUNT`- Conta i record che soddisfano i criteri specificati
- `OCCUR`- Analizza i modelli di occorrenza nei dati

Per l'operatore SPLICE, l'utilità utilizzerà un approccio multithread basato su strategie di suddivisione dei dati per garantire prestazioni ottimizzate.

I dettagli sugli operatori devono essere recuperati dalla documentazione appropriata della piattaforma legacy.

Firma

L'`ICETOOL`utilità non accetta alcun parametro, ma si basa su set di dati specifici:

- `TOOLIN`Il set di dati contiene le istruzioni di controllo che devono essere elaborate dall'utilità
- `TOOLMSG`e `DFSMSG` i set di dati non vengono utilizzati dall'`ICETOOL`utilità modernizzata per ora (ignorati)

- `IN`è il prefisso per i set di dati di input (record da elaborare)
- `OUT`è il prefisso per i set di dati di output (record risultanti dall'elaborazione)
- altri set di dati potrebbero essere referenziati dalle istruzioni di controllo nella scheda di controllo

Controlli/Gestione degli errori

Nei seguenti casi, `RuntimeException` verrà generato un messaggio con un messaggio correlato:

- Se l'operatore utilizzato in una delle istruzioni di controllo non è supportato
- Per qualsiasi operatore, se viene fornita una direttiva non supportata

Esempio di utilizzo

Esempio di ICETOOL SORT

Ecco un esempio di jcl precedente che utilizza ICETOOL per scopi di ordinamento:

- ogni istruzione di controllo dell'operatore SORT utilizza una scheda di controllo dedicata, il cui riferimento è specificato tramite la parola chiave USING
- tutte le schede di controllo sono definite dopo la `TOOLIN` definizione e sono in linea (vedere `SEL1CNTL` le `*CNTL` voci seguenti)

```
//SAMPLO52 EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG  DD SYSOUT=*
//DFSMSG   DD SYSOUT=*
//IN1      DD DSN=input(input846a.data),DISP=SHR
//            DCB=(RECFM=F,LRECL=8)
//IN2      DD DSN=input(input846b.data),DISP=SHR
//            DCB=(RECFM=F,LRECL=8)
//OUT1     DD DSN=output(out846a.txt),DISP=(,CATLG)
//            DCB=(RECFM=F,LRECL=8)
//OUT2     DD DSN=output(out846b.txt),DISP=(,CATLG)
//            DCB=(RECFM=V)
//OUT3     DD DSN=output(out846c.txt),DISP=(,CATLG)
//            DCB=(RECFM=V)
//TOOLIN   DD *
  SORT FROM(IN1) TO(OUT1) USING(SEL1)
  SORT FROM(IN2) TO(OUT1) USING(SEL2)
  SORT FROM(IN1) TO(OUT2) USING(SEL3)
```

```
  SORT FROM(IN2) TO(OUT2) USING(SEL4)
  SORT FROM(IN1) TO(OUT3) USING(SEL5)
  SORT FROM(IN2) TO(OUT3) USING(SEL6)
/*
//SEL1CNTL DD *
    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*'))
//*
//SEL2CNTL DD *
   OPTION COPY,SKIPREC=1
   OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*'))
//*
//SEL3CNTL DD *
    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '
//*
//SEL4CNTL DD *
    OPTION COPY,SKIPREC=1
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '
//*
//SEL5CNTL DD *
    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '
//*
//SEL6CNTL DD *
    OPTION COPY,SKIPREC=1
    OUTFIL BUILD=(1,7,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '
//*
```

Una volta modernizzato, lo snippet di script groovy corrispondente ha il seguente aspetto:

```
// STEP SAMPLO52 - PGM - ICETOOL**************************************************
def stepSAMPLO52(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("SAMPLO52", "ICETOOL", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
```

```
                        .systemOut("TOOLMSG")
                        .output("*")
                        .build()
                        .systemOut("DFSMSG")
                        .output("*")
                        .build()
                        .fileSystem("IN1")
                        .path("input(input846a.data)").recordSize(8)
                        .disposition("SHR")
                        .build()
                        .fileSystem("IN2")
                        .path("input(input846b.data)").recordSize(8)
                        .disposition("SHR")
                        .build()
                        .fileSystem("OUT1")
                        .path("output(out846a.txt)").recordSize(8)
                        .normalTermination("CATLG")
                        .build()
                        .fileSystem("OUT2")
                        .path("output(out846b.txt)").rdw(true)
                        .normalTermination("CATLG")
                        .build()
                        .fileSystem("OUT3")
                        .path("output(out846c.txt)").rdw(true)
                        .normalTermination("CATLG")
                        .build()
                        .fileSystem("TOOLIN")
                        .stream(
"""  SORT FROM(IN1) TO(OUT1) USING(SEL1)
  SORT FROM(IN2) TO(OUT1) USING(SEL2)
  SORT FROM(IN1) TO(OUT2) USING(SEL3)
  SORT FROM(IN2) TO(OUT2) USING(SEL4)
  SORT FROM(IN1) TO(OUT3) USING(SEL5)
  SORT FROM(IN2) TO(OUT3) USING(SEL6)               """, getEncoding())
                        .build()
                        .fileSystem("SEL1CNTL")
                        .stream(
"""    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')) """, getEncoding())
                        .build()
                        .fileSystem("SEL2CNTL")
                        .stream(
"""   OPTION COPY,SKIPREC=1
   OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*'))""", getEncoding())
```

```
                        .build()
                        .fileSystem("SEL3CNTL")
                        .stream(
"""    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '""", getEncoding())
                        .build()
                        .fileSystem("SEL4CNTL")
                        .stream(
"""    OPTION COPY,SKIPREC=1
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '""", getEncoding())
                        .build()
                        .fileSystem("SEL5CNTL")
                        .stream(
"""    OPTION COPY
    OUTFIL BUILD=(1,8,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '""", getEncoding())
                        .build()
                        .fileSystem("SEL6CNTL")
                        .stream(
"""    OPTION COPY,SKIPREC=1
    OUTFIL BUILD=(1,7,JFY=(SHIFT=LEFT,TRAIL=C'*')),
    FTOV,VLTRIM=C' '""", getEncoding())
                        .build()
                        .getFileConfigurations())
                    .withParameters(params)
                    .runProgram("ICETOOL")
                })
        }
    }
}
```

Note:

- Le schede di controllo in linea vengono utilizzate «così come sono»; non è avvenuta alcuna trasformazione rispetto alle schede di controllo precedenti
- `TOOLMSG`e `DFSMSG` sono definite nella versione modernizzata, ma verranno ignorate in fase di esecuzione
- Sia nelle versioni precedenti che in quelle modernizzate, le schede di controllo sono definite con il `CNTL` suffisso, ma nelle direttive del set di dati fanno riferimento senza il suffisso: ad esempio In,

si riferisce alla definizione del `TOOLIN` set di dati `SORT FROM(IN1) TO(OUT1) USING(SEL1)` `USING(SEL1)` `SEL1CNTL`

Esempio di copia di ICETOOL

Ecco un altro esempio di ICETOOL, che utilizza l'operatore COPY. `TOOLIN`È incluso nel frammento di script jcl:

```
//SAMPLO51 EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG  DD SYSOUT=*
//DFSMSG   DD SYSOUT=*
//IN1      DD DSN=input(input831.data),DISP=SHR
//            DCB=(RECFM=F,LRECL=12)
//OUT1     DD DSN=output(out831a.txt),DISP=OLD
//            DCB=(RECFM=F,LRECL=12)
//OUT2     DD DSN=output(out831b.txt),DISP=OLD
//            DCB=(RECFM=F,LRECL=12)
//TOOLIN   DD *
  COPY FROM(IN1) TO(OUT1,OUT2) USING(SEL1)
/*
//SEL1CNTL DD *
  OPTION COPY
  OUTFIL INCLUDE=(7,2,CH,EQ,C'10')
//*
```

Ed ecco lo snippet di script groovy modernizzato corrispondente:

```
// STEP SAMPLO51 - PGM - ICETOOL**************************************************
def stepSAMPLO51(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("SAMPLO51", "ICETOOL", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .withJobContext(jobContext)
                        .systemOut("TOOLMSG")
                        .output("*")
                        .build()
                        .systemOut("DFSMSG")
                        .output("*")
                        .build()
                        .fileSystem("IN1")
```

```
                        .path("input(input831.data)").recordSize(12)
                        .disposition("SHR")
                        .build()
                        .fileSystem("OUT1")
                        .path("output(out831a.txt)").recordSize(12)
                        .disposition("OLD")
                        .build()
                        .fileSystem("OUT2")
                        .path("output(out831b.txt)").recordSize(12)
                        .disposition("OLD")
                        .build()
                        .fileSystem("TOOLIN")
                        .stream(
"""  COPY FROM(IN1) TO(OUT1,OUT2) USING(SEL1)
  COPY FROM(IN1) TO(OUT3,OUT4)
  COPY FROM(IN1) TO(OUT4)
  COPY FROM(IN1) TO(OUT5,OUT6)                        """, getEncoding())
                        .build()
                        .fileSystem("SEL1CNTL")
                        .stream(
"""  OPTION COPY
  OUTFIL INCLUDE=(7,2,CH,EQ,C'10')""", getEncoding())
                        .build()
                        .getFileConfigurations())
                    .withParameters(params)
                    .runProgram("ICETOOL")
                })
        }
    }
}
```

MFSORT

Scopo

Questo programma di utilità ha lo scopo di imitare il comportamento dell'utilità di ordinamento denominata MFSORT presente negli ambienti Micro Focus (di solito viene richiamata dalla riga di comando o negli script in ambienti legacy).

Internamente, il programma delega le operazioni di ordinamento effettive al programma di utilità. the section called "SORT/SYNCSORT/ICEMAN"

Firma

È supportata solo la seguente sintassi precedente: `mfsort take <control card>`

La chiamata di istruzioni dirette as NON `mfsort <instructions>` è supportata.

Non richiede alcun argomento; la direttiva take viene emulata utilizzando un set di dati a cui si fa riferimento come`TAKE`, che contiene i comandi per il funzionamento di MFSORT.

Controlli/Gestione degli errori

- Se il `TAKE` set di dati è mancante o non valido, `RuntimeException` verrà generato un
- the section called "Controlli/Gestione degli errori"Si applicano anche in questo caso, data la delega da MFSORT a SORT

Esempio di utilizzo

La seguente chiamata al comando mostra un esempio di utilizzo di MFSORT:

```
mfsort take TESTSRT1.CTL
```

Ecco lo snippet di script groovy adattato e modernizzato corrispondente:

```
mpr.withFileConfigurations(new FileConfigurationUtils()
.fileSystem("TAKE")
.path("input(TESTSRT1.CTL)")
.build()
.getFileConfigurations())
.withArguments("input") // relative path for use and give files
.runProgram("MFSORT");
```

Altro/Utilità varie

Questa sezione tratta vari programmi di utilità, con scopi vari, che non possono essere collegati a categorie esistenti

Nei paragrafi seguenti, `PICTURE` deve essere intesa come la `PICTURE` parola chiave COBOL, utilizzata per descrivere il tipo di dati e i formati dei campi.

AND/CBL_OR/CBL_XOR/CBL_EQ/CBL_IMP/CBLCBL_ _NON

Scopo

Ciò si riferisce al supporto di operazioni logiche bit per bit sugli elementi di dati, presenti in alcuni dialetti COBOL (Micro Focus). Tieni presente che il refactor automatico con AWS Transform gestisce la trasformazione degli utilizzi degli operatori bit per bit dal dialetto Cobol MF a java (vedi esempi di utilizzo di seguito). Affinché il codice modernizzato funzioni correttamente, l'applicazione di utilità dovrà essere distribuita insieme.

Operatori supportati:

- Operatore unario:

Operatore unario	details
`CBL_NOT`	Complemento bit per bit (~target)

- Operatori binari:

Operatori binari	details
`CBL_AND`	AND bit per bit (sorgente e destinazione)
`CBL_OR`	OR bit per bit (fonte | destinazione)
`CBL_XOR`	OR esclusivo bit per bit (sorgente ^ destinazi one)
`CBL_EQ`	Equivalenza bit per bit (~ (source ^ target))
`CBL_IMP`	Implicazione bit per bit (~source | target)

Firma

Oltre a CBL_NOT che ha un solo operando (target), tutte le altre operazioni hanno almeno due operandi, che sono elementi di dati. Il primo operando è l'origine, il secondo operando è la

destinazione. Un argomento opzionale aggiuntivo (length) fornisce il numero di byte da elaborare (da sinistra a destra). Se la lunghezza viene omessa, il valore predefinito è il minimo delle dimensioni di origine e destinazione (in byte). Il risultato dell'operazione viene memorizzato nella destinazione. Il codice restituito dal programma è 0 (a meno che non si verifichi un'eccezione).

Controlli/Gestione degli errori

- Per ogni operatore disponibile, viene verificato il numero di argomenti richiesti. Se il numero minimo di argomenti richiesti non viene soddisfatto, `IllegalArgumentException` verrà generato un.
- Viene verificata la positività dell'argomento opzionale relativo alla lunghezza intera. Se la lunghezza fornita è negativa, nei registri verrà visualizzato un messaggio e l'operatore non verrà applicato.

Esempio di utilizzo

- Esempio di operatore unario:

Ecco un utilizzo in cobol di CBL_NOT:

```
* TARGET:       00000101
* OPERATION:     NOT
* EXPECTED:     11111010 = 64000 dec (2nd byte unchanged - all 0)
     MOVE X'0500' to TARGET
     CALL "CBL_NOT" USING TARGET BY VALUE 1.
```

e la corrispondente modernizzazione java:

```
/*
TARGET:       00000101
OPERATION:     NOT
EXPECTED:     11111010 = 64000 dec (2nd byte unchanged - all 0) */
ctx.getTarget().getTargetReference().setBytes(new byte[]{ 5, 0 });
ctrl.callSubProgram("CBL_NOT", CallBuilder.newInstance()
    .byReference(ctx.getTarget().getTargetReference())
    .byValue(1)
    .getArguments(), ctx);
```

- Esempio di operatore binario:

Vedi ad esempio il seguente codice COBOL che utilizza CBL_AND:

```
* SOURCE:       00000011
* OPERATION:     AND
* TARGET:       00000101
* EXPECTED:     00000001 = 256 dec (2nd byte unchanged - all 0)
     MOVE X'0300' to SRC
     MOVE X'0500' to TARGET
     CALL "CBL_AND" USING SRC TARGET BY VALUE 1.
```

e la sua modernizzazione in java:

```
/*
SOURCE:       00000011
OPERATION:     AND
TARGET:       00000101
EXPECTED:     00000001 = 256 dec (2nd byte unchanged - all 0) */
ctx.getSrc().getSrcReference().setBytes(new byte[]{ 3, 0 });
ctx.getTarget().getTargetReference().setBytes(new byte[]{ 5, 0 });
ctrl.callSubProgram("CBL_AND", CallBuilder.newInstance()
    .byReference(ctx.getSrc().getSrcReference())
    .byReference(ctx.getTarget().getTargetReference())
    .byValue(1)
    .getArguments(), ctx);
```

CEE3ABD

Scopo

Questo programma di utilità imita il comportamento del programma legacy con lo stesso nome, il cui ruolo consiste nel terminare un programma con un codice Abend (abnormal end) e una tempistica di pulizia opzionale. Il motore di trasformazione AWS gestisce la modernizzazione automatica delle chiamate verso CEE3 ABD. Affinché il codice modernizzato funzioni correttamente, l'applicazione di utilità dovrà essere distribuita insieme.

Il programma arresta l'unità di esecuzione corrente, utilizzando il codice abend fornito. Nei registri viene stampato un messaggio informativo che specifica che è stata richiamata l'uscita dell'utente con il codice fornito. Per ora, il parametro temporale NON viene preso in considerazione, ma viene utilizzato come argomento per ragioni di compatibilità con la modernizzazione.

Firma

Il programma accetta 0 o 2 argomenti. I due argomenti sono:

- Il codice Abend (un elemento di dati che deve essere interpretabile come un valore intero positivo)
- I tempi di pulizia (un elemento di dati che deve essere interpretabile come un valore intero positivo) sono ignorati

Quando non vengono forniti argomenti, vengono utilizzati i seguenti valori predefiniti:

- Codice Abend: 0
- Tempo di pulizia: 0

Controlli/gestione degli errori

- Verifica che vengano forniti 0 o 2 argomenti, altrimenti `IllegalArgumentException` ne verrà generato uno.
- Se vengono forniti due argomenti, verifica che: il codice abend sia compreso tra 0 e 4095 (entrambi inclusi); il timing sia compreso tra 0 e 5 (entrambi inclusi)

Qualsiasi fallimento di questi test attiverà un. `IllegalArgumentException`

Esempio di utilizzo

Ecco un esempio di utilizzo in un programma COBOL, dall'applicazione Carddemo:

```
   9999-ABEND-PROGRAM.
       DISPLAY 'ABENDING PROGRAM'
       MOVE 0 TO TIMING
       MOVE 999 TO ABCODE
       CALL 'CEE3ABD'.
```

e il corrispondente codice java modernizzato:

```
public void _9999AbendProgram(final Cbtrn03cContext ctx, final ExecutionController
 ctrl) {
    DisplayUtils.display(ctx, ctrl, LOGGER, "ABENDING PROGRAM");
    ctx.getTiming().setTiming(0);
    ctx.getAbcode().setAbcode(999);
```

```
    ctrl.callSubProgram("CEE3ABD", CallBuilder.newInstance()
        .getArguments(), ctx);
}
```

CREARE

Scopo

L'utilità CEEDATE converte un numero che rappresenta una data di Lilian (giorni dal 15 ottobre 1582) in una rappresentazione di data basata su caratteri, utilizzando un formato fornito.

Imita il comportamento dell'utilità di sistema legacy con lo stesso nome.

Firma

Data la sua natura, l'utilità CEEDATE è pensata piuttosto per essere richiamata dai programmi.

Richiede tre o quattro argomenti (l'ultimo argomento è facoltativo):

- Il primo argomento obbligatorio è un elemento di dati numerico il cui valore verrà interpretato come una data Lilian
- Il secondo argomento obbligatorio è un elemento di dati alfanumerico contenente la stringa utilizzata per la conversione della data in caratteri `PICTURE`
- Il terzo argomento obbligatorio è l'elemento di dati alfanumerico di destinazione, che contiene il risultato della conversione del primo argomento utilizzando il secondo argomento come `PICTURE`
- Il quarto argomento opzionale è un elemento di dati utilizzato per memorizzare il codice di feedback dell'utilità

Controlli/Gestione degli errori

- Se il numero di argomenti passati all'utilità non è tre o quattro, `BluageWrapperException` verrà generato un
- Se il primo argomento numerico fornito non può essere valutato correttamente come data di Lilian (oltre i limiti), verrà registrato un messaggio di errore. Il titolare del codice di feedback opzionale, se presente, riceverà un codice di feedback di gravità 3 e il messaggio numero 2512
- Se si verifica un'eccezione durante la conversione della data a causa di un dato fornito non validoPICTURE, verrà registrato un messaggio di errore. Il titolare opzionale del codice di feedback, se presente, riceverà un codice di feedback di gravità 3 e il messaggio numero 2518

- Se, per qualsiasi motivo, la conversione non può avvenire correttamente, l'elemento di dati di output verrà riempito con spazi vuoti
- Se la conversione è andata a buon fine, al contenitore opzionale del codice di feedback verrà fornito un codice di feedback di gravità 0 (e nessun messaggio)

Esempio di utilizzo

Ecco un esempio di chiamata da un programma COBOL (la struttura del detentore del codice di feedback FC contiene puntini di sospensione, così come contiene centinaia di voci di condizioni (livello 88) non mostrate qui):

```
WORKING-STORAGE SECTION.
01  LILIANS              PIC S9(9) BINARY.
01  TIMESTAMP-OUT               PIC X(80).
01  MASK.
    05  MASK-LEN         PIC S9(4) BINARY.
    05  MASK-STR.
        10  MASK-CHR     PIC X OCCURS 0 TO 256
                            DEPENDING ON MASK-LEN.
01  ROUTINE-NAMES.
    05 CEESECS-ROUTINE    PIC X(08) VALUE 'CEESECS '.
    05 CEELOCT-ROUTINE    PIC X(08) VALUE 'CEELOCT '.
    05 CEEDATE-ROUTINE    PIC X(08) VALUE 'CEEDATE '.
01  FC.
...
* lilian date for 4 June 1990
    MOVE SPACES TO MASK-STR
    MOVE 148887 TO LILIANS.
    MOVE 23                  TO MASK-LEN
    MOVE 'YYYY-MM-DD-HH:MI:SS.999' TO MASK-STR
    CALL CEEDATE-ROUTINE USING LILIANS
                               MASK
                               TIMESTAMP-OUT
                               FC.
```

CEELOCT

Scopo

L'utilità CEELOCT viene utilizzata per restituire il file locale in tre formati: date/time

- Data liliana (numero di giorni dal 14 ottobre 1582)
- Lilian secondi (il numero di secondi trascorsi dalle 00:00:00 del 14 ottobre 1582)
- Stringa di caratteri gregoriani (nel modulo) `YYYYMMDDHHMISS999`

Imita il comportamento dell'utilità di sistema legacy con lo stesso nome.

Firma

Data la sua natura, l'utilità CEELOCT è pensata piuttosto per essere richiamata dai programmi.

Richiede tre o quattro argomenti (l'ultimo argomento è facoltativo):

- Il primo argomento obbligatorio è un elemento di dati, utilizzato per memorizzare la data di Lilian
- Il secondo argomento obbligatorio è un elemento di dati, utilizzato per memorizzare i secondi di Lilian
- Il terzo argomento obbligatorio è un elemento di dati, utilizzato per memorizzare la data gregoriana utilizzando il modulo sopra indicato
- Il quarto argomento opzionale è un elemento di dati utilizzato per memorizzare il codice di feedback dell'utilità

Controlli/Gestione degli errori

- Se il numero di argomenti passati all'utilità non è tre o quattro, `BluageWrapperException` verrà generato un
- Se si verifica un'eccezione durante la gestione della conversione dal formato locale date/time a uno qualsiasi dei formati di output: il primo e il secondo argomento verranno impostati su 0 e il terzo argomento rimarrà invariato; verrà registrato un messaggio di errore; Facoltativamente, al detentore del codice di feedback verrà fornito un codice di feedback con gravità 3 e il messaggio numero 2531
- In caso di successo, tutti e tre gli argomenti verranno compilati con il contenuto corretto e il codice di feedback opzionale verrà fornito con un codice di gravità 0

Esempio di utilizzo

Ecco un frammento di codice COBOL di esempio che mostra l'utilizzo dell'utilità CEELOCT. La struttura del contenitore del codice di feedback FC non è fornita per intero in quanto contiene centinaia di voci relative alle condizioni (livello 88).

```
WORKING-STORAGE SECTION.
01  LILIANS              PIC S9(9) BINARY.
01  GREGORN              PIC X(80).
01  SECONDS              COMP-2.
01  FC.
...
01  ROUTINE-NAMES.
    05 CEESECS-ROUTINE    PIC X(08) VALUE 'CEESECS '.
    05 CEELOCT-ROUTINE    PIC X(08) VALUE 'CEELOCT '.
    05 CEEDATE-ROUTINE    PIC X(08) VALUE 'CEEDATE '.
...

    CALL CEELOCT-ROUTINE USING LILIANS
                               SECONDS
                               GREGORN
                               FC.
```

CEERAN (0)

Scopo

Il programma CEERAN0 viene chiamato per generare numeri pseudo-casuali, compresi tra 0,0 e 1,0, utilizzando un seme specificato. Si basa sull'algoritmo del metodo moltiplicativo congruenziale, che richiede un seme specificato dall'utente. L'uso di 0 come seme attiva una modalità specifica in cui il seme viene invece calcolato in base all'ora media di Greenwich (al momento dell'esecuzione del programma). Altrimenti, il seme viene usato così com'è. La sequenza pseudo-casuale è prevedibile.

Firma

Il programma CEERAN0 utilizza tre parametri:

- il seme (parametro di input), un elemento di dati che può essere interpretato come numero intero positivo (0 incluso)
- il numero casuale (parametro di output), un elemento di dati che può essere interpretato come un numero fluttuante a doppia precisione (il cui valore sarà compreso tra 0,0 e 1,0, escluso); è il risultato del programma
- il codice di feedback opzionale (parametro di output), un elemento di dati di 12 byte, utilizzato per memorizzare il feedback del programma sul calcolo dei numeri casuali

Controlli/gestione degli errori

- Se il numero di argomenti non è 2 o 3, `IllegalArgumentException` verrà generato un
- Il valore del seme deve essere compreso tra 0 e 2147483646, inclusi. Se il valore iniziale non rientra in questi limiti, verrà registrato un messaggio di errore e il codice di feedback verrà impostato sulla gravità 3 e il numero del messaggio su 2524. Il numero casuale risultante verrà impostato su -1,0 (comportamento analogico a quello precedente)
- Se il valore iniziale è impostato su 0, ma il sistema non è stato in grado di recuperare l'ora media di Greewich (per qualsiasi motivo), il calcolo verrà eseguito utilizzando il valore 1 come fallback e il codice di feedback verrà impostato sulla gravità 1 e il numero del messaggio su 2523. Il calcolo dei numeri casuali continuerà con il valore iniziale di riserva (analogico al comportamento precedente)

Esempio di utilizzo

Questo è un esempio di java che dimostra come utilizzare il programma CEERAN0, utilizzando tutti i parametri incluso il codice di feedback, con bit provenienti da diversi livelli (entità, servizio):

```
//Entity layer
public class Randomin extends RecordEntity {
    private final Group root = new Group(getData()).named("RANDOMIN");
    private final Elementary randomin = new Elementary(root,new BinaryIntegerType(4,
 true),new BigDecimal("0")).named("RANDOMIN");
    ...

public class Randomout extends RecordEntity {
    private final Group root = new Group(getData()).named("RANDOMOUT");
    private final Elementary randomout = new Elementary(root,new
 DoubleFloatingPointType(),new BigDecimal("0")).named("RANDOMOUT");
    ...
public class Returncode1 extends RecordEntity {
    private final Group root = new Group(getData()).named("RETURNCODE");
    private final Elementary returncode1 = new Elementary(root,new
 AlphanumericType(12)," ").named("RETURNCODE");
    ...

// Service layer
    CallHandler.newInstance(ctrl, ctx, ctx.getErrorContext())
        .byReference(ctx.getRandomin().getRandominReference(),
        ctx.getRandomout().getRandomoutReference(),
        ctx.getReturncode1().getReturncode1Reference())
```

```
        .call("CEERAN0");
```

FORMAGGI

Scopo

L'utilità CEESECS converte la rappresentazione di una stringa di timestamp in secondi Lilian (il numero di secondi trascorsi dalle 00:00:00 del 14 ottobre 1582).

Firma

Data la sua natura, l'utilità CEESECS è pensata piuttosto per essere richiamata dai programmi.

Richiede tre o quattro argomenti (l'ultimo argomento è facoltativo):

- Il primo argomento obbligatorio è un elemento di dati il cui valore verrà interpretato come timestamp
- Il secondo argomento obbligatorio è un elemento di dati alfanumerico contenente la `PICTURE` stringa utilizzata per specificare come interpretare il primo argomento
- Il terzo argomento obbligatorio è l'elemento di dati, che contiene il risultato della conversione del primo argomento utilizzando il secondo argomento come `PICTURE`
- Il quarto argomento opzionale è un elemento di dati utilizzato per memorizzare il codice di feedback dell'utilità

Controlli/Gestione degli errori

- Se il numero di argomenti passati all'utilità non è tre o quattro, `BluageWrapperException` verrà generato un
- Se il timestamp passato all'utilità come argomento non è valido, verrà registrato un messaggio di errore e, facoltativamente, al portatore del codice di feedback verrà fornito un codice di feedback di gravità 3 e il messaggio numero 2513
- Se l'immagine passata all'utilità come argomento non è valida, verrà registrato un messaggio di errore e, facoltativamente, al portatore del codice di feedback verrà fornito un codice di feedback di gravità 3 e il messaggio numero 2518
- Se, per qualsiasi motivo, l'output di lilian seconds non può essere calcolato, il terzo argomento (output) verrà impostato su 0

Esempio di utilizzo

Ecco un esempio di chiamata all'utilità CEESECS in un programma COBOL:

```
WORKING-STORAGE SECTION.
01  SECONDS              COMP-2.
01  TIMESTAMP-IN.
    05  TIMESTAMP-IN-LEN        PIC S9(4) BINARY.
    05  TIMESTAMP-IN-STR.
        10  TIMESTAMP-IN-CHAR   PIC X OCCURS 0 TO 256
                           DEPENDING ON TIMESTAMP-IN-LEN.
01  MASK.
    05  MASK-LEN         PIC S9(4) BINARY.
    05  MASK-STR.
        10  MASK-CHR     PIC X OCCURS 0 TO 256
                           DEPENDING ON MASK-LEN.
01  FC.
...
01  ROUTINE-NAMES.
    05 CEESECS-ROUTINE    PIC X(08) VALUE 'CEESECS '.
    05 CEELOCT-ROUTINE    PIC X(08) VALUE 'CEELOCT '.
    05 CEEDATE-ROUTINE    PIC X(08) VALUE 'CEEDATE '.
...
...
* date for lilian second 12,799,191,601.123
    MOVE '1988-5-16-19:00:01.123' TO TIMESTAMP-IN-STR
    MOVE 23                   TO MASK-LEN
    MOVE 'YYYY-MM-DD-HH:MI:SS.999' TO MASK-STR
    CALL CEESECS-ROUTINE USING TIMESTAMP-IN
                               MASK
                               SECONDS
                               FC.
```

ILBOABN0

Scopo

Lo scopo del programma ILBOABN0 è interrompere l'unità di esecuzione corrente in modo controllato, utilizzando un codice abend (fine anomalo) fornito dall'utente. Spesso utilizzato nella gestione degli errori nei programmi dedicati.

L'interruzione dell'unità di funzionamento corrente avviene lanciando un. `StopRunUnitException`

Firma

Il programma ILBOABN0 accetta un singolo argomento obbligatorio che è un elemento di dati contenente il codice abend (che deve essere interpretabile come un numero intero).

Controlli/Gestione degli errori

Durante il lancio di `StopRunUnitException` to interrupt l'esecuzione corrente dell'unità di esecuzione, il programma imposterà il codice di ritorno sul valore fornito come primo argomento. Inoltre, verrà registrato un messaggio informativo.

Esempio di utilizzo

Ecco un esempio di utilizzo in java del programma ILBOABN0, derivante da una modernizzazione COBOL tramite AWS Transform:

```
   77  WS-ABND-CODE          COMP   PIC S9(4)      VALUE +1234.
...
...

   1970-ABNDIT.
       CALL 'ILBOABN0' USING WS-ABND-CODE.
```

e la corrispondente modernizzazione java:

```
//Entity layer
private final Group root = new Group(getData());
private final Elementary wsAbndCode = new Elementary(root,new BinaryType(4, 0, "STD",
 false, false, true),Short.valueOf("1234"));
...

//Service layer
@Override
public void _1970Abndit(final MyPgmContext ctx, final ExecutionController ctrl) {
   ctrl.callSubProgram("ILBOABN0", CallBuilder.newInstance()
       .byReference(ctx.getWsAbndCode().getWsAbndCodeReference())
       .getArguments(), ctx);
```

Configurare AWS Transform for mainframe Runtime

Questa sezione spiega i passaggi per configurare AWS Transform for mainframe Runtime sulla tua AWS infrastruttura. Prima di configurare AWS Transform for mainframe Runtime for Applications,

è necessario comprendere i prerequisiti, le regioni e i bucket e la configurazione degli CloudWatch allarmi necessari per configurare e gestire l'ambiente di runtime.

Argomenti

- AWS Prerequisiti di Transform for mainframe Runtime
- Onboarding AWS Transform per mainframe Runtime
- Requisiti di configurazione dell'infrastruttura per AWS Transform for mainframe Runtime
- AWS Transform for mainframe Runtime artefact
- Implementa AWS Transform for mainframe Runtime su Amazon EC2
- Implementa AWS Transform for mainframe Runtime su container su Amazon ECS e Amazon EKS
- Prova l' PlanetsDemo applicazione

AWS Prerequisiti di Transform for mainframe Runtime

AWS Transform for mainframe Runtime è disponibile in diverse versioni di the section called "AWS Note sulla versione di Transform for mainframe" 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 vostre esigenze, contattate il vostro responsabile della distribuzione di AWS Transform for mainframe.

Prima di iniziare a utilizzare AWS Transform for mainframe Runtime, procedi come segue:

- Assicuratevi di avere un AWS account.
- Assicurati di avere un'applicazione modernizzata rifattorizzata con AWS Transform for mainframe.
- Scegli una AWS regione e una delle opzioni di elaborazione supportate per Transform for AWS mainframe Runtime.
- Scegli la versione di AWS Transform for mainframe Runtime che desideri utilizzare.
- Rivedi the section called "Requisiti di configurazione dell'infrastruttura" e convalida i componenti aggiuntivi necessari per eseguire AWS Transform for mainframe Runtime.

> **Note**
>
> Se desideri testare le funzionalità di AWS Transform for mainframe Runtime, puoi utilizzare l'applicazione demo`Planets Demo`, che puoi scaricare da -v1.zip. PlanetsDemo

Onboarding AWS Transform per mainframe Runtime

> **Note**
>
> L'accesso a AWS Transform for mainframe Runtime è passato dalle richieste basate su ticket al download diretto tramite AWS Transform for mainframe refactor. Abbiamo introdotto un modo nuovo e semplificato per accedere ai AWS Transform for mainframe prodotti tramite la funzionalità AWS Transform for mainframeToolbox in AWS Transform for mainframe refactor.

Per iniziare, crea un Supporto AWScaso per richiedere l'onboarding per accedere a Transform for mainframe Runtime. AWS Includi nella richiesta il tuo Account AWS ID, la AWS regione che desideri utilizzare e una scelta di elaborazione e la versione AWS Transform for mainframe Runtime. Se non sei sicuro di quale versione ti serve, contatta il tuo responsabile della distribuzione di AWS Transform for mainframe. Se disponi già del codice sorgente dell'applicazione generato dagli strumenti AWS Mainframe Modernization Refactoring, annota il valore del `gapwalk.version` tag in un `pom.xml` file all'interno della tua base di codice modernizzata.

> **Note**
>
> AWS Transform for mainframe Runtime è disponibile in due varianti principali: versioni pre-release Alpha e Releases. Per determinare quale versione utilizzarethe section called "AWS Transform per il controllo delle versioni mainframe", consultate o contattate il vostro responsabile della distribuzione di AWS Transform for mainframe.

Regioni e bucket per AWS Transform for mainframe Runtime

Archiviamo gli artefatti AWS Transform for mainframe Runtime in diversi bucket Amazon S3 per regione e per scelta di elaborazione. Per accedere al bucket per for AWS Transform Regione AWS for mainframe Runtime, usa il nome elencato nella tabella seguente.

Regione AWS	Secchio di rilascio	Bucket Alpha pre-release
Stati Uniti orientali (Ohio)	aws-bluage-runtime-artifacts-055777665268-us-east-2	aws-bluage-runtime-artifacts-dev-055777665268-us-east-2

Regione AWS	Secchio di rilascio	Bucket Alpha pre-release
Stati Uniti orientali (Virginia settentrionale)	aws-bluage-runtime-artifacts-139023371234-us-est-1	aws-bluage-runtime-artifacts-dev-139023371234-us-east-1
Stati Uniti occidentali (California settentrionale)	aws-bluage-runtime-artifacts-788454048782-us-west-1	aws-bluage-runtime-artifacts-dev-788454048782-us-west-1
Stati Uniti occidentali (Oregon)	aws-bluage-runtime-artifacts-836771190483-us-west-2	aws-bluage-runtime-artifacts-dev-836771190483-us-west-2
Canada (Centrale)	aws-bluage-runtime-artifacts-637423580979-ca-central-1	aws-bluage-runtime-artifacts-dev-637423580979-ca-central-1
Europa (Irlanda)	aws-bluage-runtime-artifacts-925278190477-eu-west-1	aws-bluage-runtime-artifacts-dev-925278190477-eu-west-1
Europa (Londra)	aws-bluage-runtime-artifacts-767397831990-eu-west-1	aws-bluage-runtime-artifacts-dev-767397831990-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
Europa (Stoccolma)	aws-bluage-runtime-artifacts-654654484534-eu-nord-1	aws-bluage-runtime-artifacts-dev-654654484534-eu-nord-1
Europa (Milano)	aws-bluage-runtime-artifacts-654654328338-eu-sud-1	aws-bluage-runtime-artifacts-dev-654654328338-eu-sud-1
Europa (Spagna)	aws-bluage-runtime-artifacts-905417994954-eu-sud-2	aws-bluage-runtime-artifacts-dev-905417994954-eu-sud-2
Sud America (San Paolo)	aws-bluage-runtime-artifacts-737536804457-sa-est-1	aws-bluage-runtime-artifacts-dev-737536804457-sa-est-1

Regione AWS	Secchio di rilascio	Bucket Alpha pre-release
Asia Pacifico (Tokyo)	aws-bluage-runtime-artifacts-445578176276-ap-nord-est-1	aws-bluage-runtime-artifacts-dev-445578176276-ap-nord-est-1
Asia Pacifico (Seul)	aws-bluage-runtime-artifacts-381492221498-ap-nordest-2	aws-bluage-runtime-artifacts-dev-381492221498-ap-nordest-2
Asia Pacifico (Osaka)	aws-bluage-runtime-artifacts-905418229615-ap-nord-est-3	aws-bluage-runtime-artifacts-dev-905418229615-ap-nord-est-3
Asia Pacifico (Singapore)	aws-bluage-runtime-artifacts-767397774613-ap-sudest-1	aws-bluage-runtime-artifacts-dev-767397774613-ap-sudest-1
Asia Pacifico (Sydney)	aws-bluage-runtime-artifacts-726160321909-ap-sutheast-2	aws-bluage-runtime-artifacts-dev-726160321909-ap-sudest-2
Asia Pacifico (Mumbai)	aws-bluage-runtime-artifacts-905418353577-ap-sud-1	aws-bluage-runtime-artifacts-dev-905418353577-ap-sud-1
Africa (Città del Capo)	aws-bluage-runtime-artifacts-992382777663-af-sud-1	aws-bluage-runtime-artifacts-dev-992382777663-af-sud-1
Israele (Tel Aviv)	aws-bluage-runtime-artifacts-471112516508-il-central-1	aws-bluage-runtime-artifacts-dev-471112516508-il-centrale-1

Utilizzo di per elencare il contenuto del bucket AWS CLI

Dopo l'onboarding, puoi elencare il contenuto del bucket eseguendo il seguente AWS CLI 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 Transform for mainframe Runtime, come la seguente per un bucket Release:

```
PRE 4.0.0/
PRE 4.10.0/
```

Oppure quanto segue per un bucket di compilazione:

```
PRE 4.11.0/
PRE 4.13.0/
```

Ti consigliamo 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 sono riportati due esempi.

Per un bucket di rilascio:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.10.0/
Framework/
```

Il comando restituisce un elenco di file zip, ad esempio:

```
2025-04-08 16:11:19  152040176 aws-bluage-runtime-4.10.0.zip
2025-04-08 16:11:52  176518889 aws-bluage-webapps-4.10.0.zip
```

Per un bucket Alpha pre-release:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-dev-139023371234-us-
east-1/4.11.0/Framework/
```

Il comando restituisce un elenco di file zip, ad esempio:

```
2025-04-09 20:23:34  152304534 aws-bluage-runtime-4.11.0.zip
2025-04-09 20:24:07  176262381 aws-bluage-webapps-4.11.0.zip
```

Scarica il framework

Puoi scaricare il framework, ad esempio, per aggiornare la versione AWS Transform for mainframe 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-1/4.0.0/Framework/ . --recursive`

Questo comando produce il seguente output:

```
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/Framework/aws-
bluage-webapps-4.0.0.zip to ./aws-bluage-webapps-4.0.0.zip
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/Framework/aws-
bluage-runtime-4.0.0.zip to ./aws-bluage-runtime-4.0.0.zip
```

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 152040176 Apr  8 16:11 aws-bluage-
runtime-4.0.0.zip
-rw-rw-r-- 1 cloudshell-user cloudshell-user 176518889 Apr  8 16:11 aws-bluage-
webapps-4.0.0.zip
```

> **Note**
>
> L'accesso agli artefatti potrebbe essere temporaneamente interrotto e le versioni potrebbero essere rimosse per motivi di sicurezza. Ti consigliamo vivamente di archiviare gli artefatti

che utilizzi nel tuo account. La versione locale deve essere utilizzata come riferimento nelle architetture interne.

Requisiti di configurazione dell'infrastruttura per AWS Transform for mainframe Runtime

Questo argomento descrive la configurazione minima dell'infrastruttura richiesta per eseguire AWS Transform for mainframe Runtime. Le seguenti procedure descrivono come configurare AWS Transform for mainframe Runtime sul computer prescelto per distribuire un'applicazione modernizzata su Transform for mainframe Runtime. AWS Le risorse che crei devono trovarsi in un Amazon VPC con una sottorete dedicata al dominio dell'applicazione.

Argomenti

- Requisiti di infrastruttura
- Esecuzione di AWS Transform for mainframe Runtime su Amazon EC2
- Esecuzione di AWS Transform for mainframe Runtime su Amazon ECS su Amazon EC2
- Esecuzione di AWS Transform for mainframe Runtime su Amazon EKS su Amazon EC2
- Esecuzione di AWS Transform for mainframe Runtime su Amazon ECS gestito da AWS Fargate

Requisiti di infrastruttura

Creare un gruppo di sicurezza

Se hai intenzione di lavorare su istanze Amazon EC2 su Amazon EKS, salta questa procedura perché il processo di creazione del cluster Amazon EKS crea un gruppo di sicurezza per tuo conto. Utilizza quel gruppo di sicurezza nelle seguenti procedure invece di crearne uno nuovo.

1. Apri la 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).

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, a seconda della scelta di elaborazione, scegli una delle seguenti opzioni:
   - EC2 (per Amazon EC2 e Amazon EKS su Amazon EC2)
   - Elastic Container Service e quindi ruolo EC2 per Elastic Container Service (per Amazon ECS su Amazon EC2)
   - Elastic Container Service e poi Elastic Container Service Task (per Amazon ECS gestito da Fargate)
5. Scegli Next (Successivo).
6. Immetti un nome del ruolo, quindi scegli Crea ruolo.

Esecuzione di AWS Transform for mainframe Runtime su Amazon EC2

Per creare un'istanza Amazon EC2, utilizza i seguenti passaggi.

Creazione di un'istanza Amazon EC2

1. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Scegliere Launch Instance (Avvia istanza).
3. Per Tipo di istanza, scegli un tipo di istanza EC2 disponibile.
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).

Installa l'applicazione sull'istanza Amazon EC2

1. Quando lo stato dell'istanza Amazon EC2 passa a Running, connettiti all'istanza.
2. Installa i seguenti componenti software sull'istanza (fai riferimento alle versioni menzionate nell'istanzathe section called "AWS Note sulla versione di Transform for mainframe"):

   - Java Runtime Environment (JRE).
   - Apache Tomcat.
   - AWS Transform for mainframe Runtime (su Amazon EC2). Installa il runtime AWS Transform for mainframe nella radice della cartella di installazione di Apache Tomcat (alcuni file verranno aggiunti mentre altri verranno sovrascritti).

   Per installare le webapp aggiuntive fornite insieme all'archivio AWS Transform for mainframe Runtime, configura un'istanza secondaria del server Apache Tomcat e decomprimi l'archivio delle webapps in quella posizione. Per istruzioni dettagliate, vedi the section called "AWS Transform for mainframe Runtime artefact".

Esecuzione di AWS Transform for mainframe Runtime su Amazon ECS su Amazon EC2

1. Crea un cluster Amazon ECS, con istanze Amazon EC2 come infrastruttura sottostante. Consulta la sezione Guida introduttiva a Windows su Amazon EC2 nella Guida per gli sviluppatori di Amazon Elastic Container Service.
2. Specificate il ruolo IAM che avete creato nei passaggi precedenti.
3. Scegli un tipo di istanza EC2.
4. Nelle impostazioni di rete per le istanze Amazon EC2, scegli il gruppo di sicurezza creato nei passaggi precedenti.

Esecuzione di AWS Transform for mainframe Runtime su Amazon EKS su Amazon EC2

1. Crea un cluster Amazon EKS. Consulta Creazione di un cluster Amazon EKS nella Guida per l'utente di Amazon EKS.
2. Come accennato in precedenza, viene creato un gruppo di sicurezza per tuo conto.
3. Crea un gruppo di nodi. Specificate il ruolo IAM che avete creato nei passaggi precedenti.
4. Scegli un tipo di istanza EC2.
5. Amazon EKS assegnerà automaticamente il gruppo di sicurezza alle istanze Amazon EC2 generate.

Esecuzione di AWS Transform for mainframe Runtime su Amazon ECS gestito da AWS Fargate

Crea un cluster Amazon ECS con AWS Fargate (serverless) come infrastruttura sottostante. Vedi Guida introduttiva a Fargate nella Amazon Elastic Container Service Developer Guide.

AWS Transform for mainframe Runtime artefact

AWS Gli artefatti di Transform for mainframe Runtime sono i componenti per la distribuzione e l'esecuzione di applicazioni modernizzate. Questo documento descrive i diversi tipi di artefatti disponibili, le relative posizioni di archiviazione e come accedervi.

AWS Artefatti di Transform for mainframe Runtime

Contenuto degli artefatti

In entrambi i bucket Release e Pre-release, troverai:

gapwalk-x.y.z.zip

Questo archivio, in cui x.y.z rappresenta il numero di versione (vedi AWS Transform for mainframe versioning), contiene i componenti principali di Transform for mainframe Runtime essenziali per l'esecuzione di AWS Transform for mainframe, tra cui: AWS

Artifact	Description	Tipo	Cartella di distribuzione
gapwalk-application-x.y.z.war	AWS Transform for mainframe applicazione web principale Gapwalk	guerra	webapp
*.jar	AWS Transform for mainframe Librerie condivise di Runtime Framework (Gapwalk)	Barattoli	shared
gapwalk-*-x.y.z.war	AWS Transform for mainframe applicazioni web aggiuntive (utility-pgm, supporto gerarchico,...)	Guerra	webapps-extra
bac-x.y.z.war/ jac-x.y.z.war	AWS Transform for mainframe Console di amministrazione (non autonome)	Guerra	webapp-console

aws-bluage-webapps-x.y.z.zip

Questo archivio, in cui x.y.z segue lo stesso schema di versioni di cui sopra, include Administration Consoles (standalone) AWS Transform for mainframe

- File WAR BAC (Blusamconsole), utilizzato per monitorare il database. Blusam
- File WAR JAC (console JICS), utilizzato per monitorare il database JICS.
- Librerie di supporto necessarie.

Verifica della firma del pacchetto

I file zip rilasciati vengono forniti come archivi ZIP con firma digitale per garantire sicurezza e autenticità. Le firme digitali aiutano a verificare che i file scaricati siano autentici, inalterati e rilasciati ufficialmente dalla nostra organizzazione. Ciò impedisce la manomissione e protegge da malware o file compromessi che potrebbero essere distribuiti in modo malevolo.

Per verificare la firma del file ZIP prima di utilizzarlo, puoi usare il seguente comando:

```
jarsigner -verify -certs -verbose aws-bluage-runtime-x.y.z.zip
```

Ecco l'esempio di risultato per un file certificato:

```
...
jar verified.
```

Per i dettagli su come vengono affrontate le vulnerabilità di sicurezza, consulta AWS Mainframe Modernization Refactor con panoramica della versione. AWS Transform for mainframe

> **Note**
>
> Facciamo del nostro meglio per rilasciare i nostri prodotti senza CVEs, ma in seguito potrebbero apparire nuovi CVEs prodotti.

Implementa AWS Transform for mainframe Runtime su Amazon EC2

Puoi scoprire come configurare AWS Transform for mainframe Runtime su Amazon EC2, come aggiornare la versione di runtime, come monitorare la distribuzione utilizzando gli allarmi CloudWatch Amazon e come aggiungere dipendenze concesse in licenza con gli argomenti di questa sezione. Queste istruzioni sono applicabili quando crei istanze Amazon EC2 e quando utilizzi Amazon ECS su Amazon EC2 o Amazon EKS su Amazon EC2.

A partire dalla versione alpha 5.60.0, la distribuzione AWS Transform for mainframe Runtime include una cartella di distribuzione con un file README.md completo. Questa guida fornisce istruzioni per la distribuzione di step-by-step Amazon EC2, opzioni di configurazione e suggerimenti per la risoluzione dei problemi.

Argomenti

- Configurare AWS Transform for mainframe Runtime su Amazon EC2
- Esegui l'upgrade di AWS Transform for mainframe Runtime su Amazon EC2
- Configurazione degli allarmi Amazon AWS Transform for mainframe Runtime (su Amazon EC2) CloudWatch
- Configura le dipendenze concesse in licenza in AWS Transform for mainframe Runtime su Amazon EC2

Configurare AWS Transform for mainframe Runtime su Amazon EC2

Questo argomento spiega come configurare e distribuire l'applicazione di PlanetsDemo esempio utilizzando AWS Transform for mainframe Runtime su Amazon EC2.

Argomenti

- Prerequisiti
- Configurazione
- Verifica l'applicazione distribuita

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 Transform for mainframe Runtime" e. the section called "Onboarding AWS Transform per mainframe Runtime"
- Crea un'istanza Amazon EC2 utilizzando uno dei tipi di istanza supportati. 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.

> **Note**
>
> In questa guida, si presume che il percorso di installazione di Tomcat sia. `/m2-anywhere/tomcat-gapwalk/velocity` Assicuratevi di utilizzare questo percorso quando seguite le istruzioni riportate di seguito o adattate le seguenti istruzioni al percorso di vostra scelta.

- Scarica ed estrai AWS Transform for mainframe Runtime (su Amazon EC2) direttamente `/m2-anywhere/tomcat-gapwalk/velocity` nella directory velocity (la cartella di installazione di Tomcat). Per istruzioni su come recuperare gli artefatti di AWS Transform for mainframe Runtime, incluse informazioni sullo storage, l'accesso e il contenuto, consulta. the section called "AWS Transform for mainframe Runtime artefact"
- Scaricate l'archivio dell'applicazione. PlanetsDemo
- Decomprimi l'archivio e carica l'applicazione in un bucket Amazon S3 a tua scelta.

- Crea un database Amazon Aurora PostgreSQL per JICS. AWS Transform for mainframe Runtime eseguirà automaticamente lo script al `PlanetsDemo-v1/jics/sql/initJics.sql` primo avvio. 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. 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 `/m2-anywhere/tomcat-gapwalk/velocity/webapps`.
3. Copia i PlanetsDemo file binari disponibili nella `PlanetsDemo-v1/webapps/` cartella 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/` `/m2-anywhere/tomcat-gapwalk/velocity/config/`
5. Fornisci le informazioni di connessione per il database che hai creato come parte dei prerequisiti nel seguente frammento del file. `/m2-anywhere/tomcat-gapwalk/velocity/config/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.

```
/m2-anywhere/tomcat-gapwalk/velocity/startup.sh

tail -f /m2-anywhere/tomcat-gapwalk/velocity/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.

Verifica l'applicazione distribuita

Per un esempio di come testare l' PlanetsDemo applicazione, vederethe section called "Prova l' PlanetsDemo applicazione".

Esegui l'upgrade di AWS Transform for mainframe Runtime su Amazon EC2

Questa guida descrive come aggiornare AWS Transform for mainframe Runtime su Amazon EC2.

Argomenti

- Prerequisiti
- Esegui l'upgrade di AWS Transform for mainframe Runtime nell'istanza Amazon EC2
- Aggiorna il runtime AWS Transform for mainframe in un contenitore

Prerequisiti

Prima di iniziare, assicurati di soddisfare i seguenti prerequisiti.

- Per verificare se esistono istruzioni specifiche per la versione in uso, consultathe section called "Aggiornamento di Transform per mainframe AWS ".
- Completa the section called "AWS Prerequisiti di Transform for mainframe Runtime" ethe section called "Onboarding AWS Transform per mainframe Runtime".

- Assicurati di avere un'istanza Amazon EC2 con un'installazione esistente di AWS Transform for mainframe Runtime. 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 Transform for mainframe Runtime a cui desideri eseguire l'aggiornamento. Il framework è composto da due file di archivio: `gapwalk-x.y.z.zip` `eaws-bluage-webapps-x.y.z.zip`. Per ulteriori informazioni, consulta the section called "AWS Transform for mainframe Runtime artefact".

Esegui l'upgrade di AWS Transform for mainframe Runtime nell'istanza Amazon EC2

Completa i seguenti passaggi per aggiornare AWS Transform for mainframe Runtime.

1. Connettiti alla tua istanza Amazon EC2 e cambia l'utente in su eseguendo il comando seguente.

   ```
   sudo su
   ```

   È necessario il privilegio di superutente per eseguire i comandi in questo tutorial.

2. Per estrarre i file binari, usa i seguenti comandi. Esegui i comandi in ogni cartella.

   ```
   unzip gapwalk-x.y.z.zip
   unzip aws-bluage-webapps-x.y.z.zip
   ```

3. Arrestare i servizi Apache Tomcat utilizzando i seguenti comandi.

   ```
   systemctl stop tomcat.service
   systemctl stop tomcat-webapps.service
   ```

4. Sostituisci il contenuto di `<your-tomcat-path>/shared/` con il contenuto di. `gapwalk-x.y.z/shared/`
5. Sostituisci `<your-tomcat-path>/webapps/gapwalk-application.war` con `gapwalk-x.y.z/webapps/gapwalk-application.war`.
6. Sostituisci i file di guerra in`<your-tomcat-path>/webapps/`, vale a dire `bac.war` `ejac.war`, con gli stessi file di`aws-bluage-webapps-x.y.z/velocity/webapps/`.
7. Avvia i servizi Apache Tomcat eseguendo i seguenti comandi.

```
systemctl start tomcat.service
systemctl start tomcat-webapps.service
```

8. 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 di AWS Transform for mainframe è stato aggiornato correttamente.

Aggiorna il runtime AWS Transform for mainframe in un contenitore

Completa i seguenti passaggi per aggiornare AWS Transform for mainframe Runtime.

1. Ricostruisci l'immagine Docker con la versione AWS Transform for mainframe Runtime desiderata. Per istruzioni, consulta the section called "Configurare AWS Transform for mainframe Runtime su Amazon EC2".
2. Invia la tua immagine Docker al tuo repository Amazon ECR.
3. Interrompi e riavvia il servizio Amazon ECS o Amazon EKS.
4. Controllo dei log.

AWS Transform for mainframe Runtime è stato aggiornato correttamente.

Configurazione degli allarmi Amazon AWS Transform for mainframe Runtime (su Amazon EC2) CloudWatch

Puoi CloudWatch configurare la ricezione del registro dell'applicazione e aggiungere un allarme per avvisarti di possibili errori. Ciò consente di avere notifiche più visibili ogni volta che le applicazioni distribuite incontrano eccezioni. Le seguenti sezioni aiutano a comprendere e conoscere la configurazione della CloudWatch registrazione e della configurazione degli allarmi.

Abilita la registrazione CloudWatch

Per abilitare CloudWatch la registrazione, create un `logback-cloudwatch.xml` file nella cartella config utilizzando il modello di configurazione fornito nella sezione seguente. Quindi, fate riferimento a questo file nel `application-main.yml` file decommentando le seguenti righe:

```
# logging:
#  config: classpath:logback-cloudwatch.xml
```

Dopo aver rimosso il commento da queste righe, CloudWatch la registrazione verrà attivata e potrà essere ulteriormente configurata come spiegato nelle sezioni seguenti.

Configurazione della registrazione CloudWatch

Utilizzate il seguente modello per creare il `logback-cloudwatch.xml` file:

```
<?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.netfective.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 è al di fuori <appender name="cloudwatch"/> dell'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.
- <maxqueuewaittimemillis/>: 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 tale scopo, è possibile aggiungere una metrica relativa alla stringa «Error C5001" nei log, che evidenzia gli errori nella connessione al sistema di controllo Transform for mainframe. AWS È quindi possibile definire un allarme che reagisca a questa metrica.

Configura le dipendenze concesse in licenza in AWS Transform for mainframe Runtime su Amazon EC2

Questa guida descrive come configurare dipendenze con licenza aggiuntive da utilizzare con AWS Transform for mainframe Runtime su Amazon EC2.

Argomenti

- Prerequisiti
- Fasi di installazione

Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Completa the section called "AWS Prerequisiti di Transform for mainframe Runtime" e. the section called "Onboarding AWS Transform per mainframe Runtime"
- Assicurati di avere un'istanza Amazon EC2 con AWS Transform for mainframe Runtime (su Amazon EC2) installata. 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 Transform for mainframe Runtime (su Amazon EC2) con la versione ojdbc11-23.3.0.23.09.jar, ma una versione più recente potrebbe essere compatibile.

Connessione IBM MQ

Fornisci un client IBM MQ. Abbiamo testato la funzionalità AWS Transform for mainframe Runtime (su Amazon EC2) con la versione com.ibm.mq.jakarta.client-9.3.4.1.jar, ma una versione più recente potrebbe essere compatibile.

Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15 a 18-1.76.jar
- bcutil-jdk15 a 18-1.76.jar

File della stampante DDS

Fornisci la libreria di report Jasper (munity.jaspersoft)https://com. com/download-jaspersoft/community-edizione). Abbiamo testato la funzionalità AWS Transform for mainframe 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
- commons-beanutils-1.9.4.jar
- commons-collections-3.2.2.jar

Fasi di installazione

Per installare le dipendenze, completa i passaggi seguenti.

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.

   > **Note**
   >
   > Per le webapp JAC e BAC con database ospitati da Oracle, assicurati di includere il driver del database Oracle (ojdbc) 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.

Implementa AWS Transform for mainframe Runtime su container su Amazon ECS e Amazon EKS

Puoi utilizzare gli argomenti di questa sezione per scoprire come configurare AWS Transform for mainframe Runtime sui container per distribuirlo su Amazon ECS (gestito da Amazon EC2 o) e Amazon EKS gestito da Amazon EC2 AWS Fargate, come aggiornare la versione di runtime, come monitorare la distribuzione utilizzando Amazon CloudWatch alarms e come aggiungere dipendenze concesse in licenza.

> **Note**
>
> Non è compatibile con Amazon EKS gestito da AWS Fargate.

Argomenti

- Configurare AWS Transform for mainframe Runtime su container
- Aggiorna AWS Transform for mainframe Runtime su container
- Configura gli CloudWatch allarmi Amazon per AWS Transform for mainframe Runtime su container
- Configura le dipendenze concesse in licenza in AWS Transform for mainframe Runtime on container

Configurare AWS Transform for mainframe Runtime su container

Questo argomento spiega come configurare e distribuire l'applicazione di PlanetsDemo esempio utilizzando AWS Transform for mainframe Runtime su un contenitore docker.

AWS Transform for mainframe Runtime on container è disponibile per Amazon ECS gestito da Amazon EC2, Amazon ECS gestito da e Amazon AWS Fargate EKS gestito da Amazon EC2. Non è compatibile con Amazon EKS gestito da AWS Fargate.

Argomenti

- Prerequisiti
- Configurazione
- Testa l'applicazione distribuita

Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Configura il AWS CLI seguendo i passaggi descritti in Configurazione della CLI di AWS.
- Completa the section called "AWS Prerequisiti di Transform for mainframe Runtime" e. the section called "Onboarding AWS Transform per mainframe Runtime"
- Scarica i file binari di AWS Transform for mainframe Runtime. Per istruzioni, consulta the section called "Onboarding AWS Transform per mainframe Runtime".
- Scarica i binari di Apache Tomcat.
- 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, seguito da AWS Transform for mainframe Runtime (per Amazon ECS managed by AWS Fargate) estratto nella radice della directory di installazione di Apache Tomcat, 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-runtime-4.x.x.zip /usr/local/velocity/installation/gapwalk/
gapwalk.zip

# run relevant installation scripts
RUN ./install-gapwalk.sh
RUN ./install-app.sh

EXPOSE 8080
EXPOSE 8081
# ...

WORKDIR /bluage/tomcat.gapwalk/velocity
# Run Command to start Tomcat server
CMD ["sh", "-c", "sudo bin/catalina.sh run"]
```

Di seguito sono riportati i contenuti di `install-gapwalk.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 AWS Transform for mainframe runtime and tomcat archives to temporary
 extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk.zip ${TEMP_DIR}
sudo cp /usr/local/velocity/installation/tomcat.tar.gz ${TEMP_DIR}
# Create velocity dir
mkdir -p /bluage/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-10.x.x/* /bluage/tomcat.gapwalk/velocity
# Remove default webapps of Tomcat
rm -f /bluage-on-fargate/tomcat.gapwalk/velocity/webapps/*
# Extract AWS Transform for mainframe runtime at velocity dir
unzip ${TEMP_DIR}/gapwalk.zip -d /bluage/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/
```

4. 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 :
```

5. Crea e trasferisci l'immagine nel tuo repository Amazon ECR. Per istruzioni, consulta Pushing a Docker image nella Amazon Elastic Container Registry User Guide. Quindi, a seconda della situazione, crea un pod Amazon EKS o una definizione di attività Amazon ECS utilizzando l'immagine Amazon ECR e distribuiscila nel tuo cluster. Ad esempio, sulla creazione di questi, consulta Creazione di una definizione di attività utilizzando la console nella Guida per gli sviluppatori di Amazon Elastic Container Service (Amazon ECS) e Distribuire un'applicazione di esempio nella Amazon EKS User Guide.
6. In particolare, per Amazon ECS gestito per AWS Fargate caso, quando crei la definizione del task, usa il ruolo IAM che hai creato come parte della configurazione iniziale dell'infrastruttura. Quindi, durante la creazione del servizio, espandi la sezione Rete e configura il VPC, le sottoreti e il gruppo di sicurezza che hai creato come parte della configurazione iniziale dell'infrastruttura. Vedi, Requisiti di configurazione dell'infrastruttura per AWS Transform for mainframe Runtime.

Testa l'applicazione distribuita

Per un esempio di come testare l' PlanetsDemo applicazione, vederethe section called "Prova l' PlanetsDemo applicazione".

Aggiorna AWS Transform for mainframe Runtime su container

Questa guida descrive come aggiornare AWS Transform for mainframe Runtime su container. A tale scopo, è necessario innanzitutto completare alcuni prerequisiti e quindi utilizzare l'immagine Docker per aggiornare AWS Transform for mainframe Runtime.

Argomenti

- Prerequisiti
- Aggiorna AWS Transform for mainframe Runtime

Prerequisiti

Prima di iniziare, assicurati di soddisfare i seguenti prerequisiti.

- Completa the section called "AWS Prerequisiti di Transform for mainframe Runtime" e. the section called "Onboarding AWS Transform per mainframe Runtime"
- Scaricate la versione di AWS Transform for mainframe Runtime a cui desiderate effettuare l'aggiornamento. Per ulteriori informazioni, consulta the section called "Onboarding AWS Transform per mainframe Runtime". Il framework è composto da due file binari: `aws-bluage-runtime-x.y.z.zip` e`aws-bluage-webapps-x.y.z.zip`.

Aggiorna AWS Transform for mainframe Runtime

Completa i seguenti passaggi per aggiornare AWS Transform for mainframe Runtime.

1. Ricostruisci l'immagine Docker con la versione AWS Transform for mainframe Runtime desiderata. Per istruzioni, consulta the section called "Configura AWS Transform for mainframe Runtime su un contenitore".
2. Invia la tua immagine Docker al tuo repository Amazon ECR.
3. Interrompi e riavvia il servizio Amazon ECS o Amazon EKS.
4. Verifica i log.

Il runtime di AWS Transform for mainframe è stato aggiornato correttamente.

Configura gli CloudWatch allarmi Amazon per AWS Transform for mainframe Runtime su container

Puoi configurare la configurazione CloudWatch per avere notifiche più visibili ogni volta che le applicazioni distribuite incontrano eccezioni. Questo ti aiuta a monitorare il registro dell'applicazione reindirizzato e CloudWatch ad aggiungere un allarme per avvisarti 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, è possibile impostare allarmi proattivi per avvisi di utilizzo direttamente durante la creazione del cluster, in modo da ricevere notifiche quando si verificano errori. Per evidenziare gli errori nella connessione al sistema di controllo AWS Transform for mainframe, aggiungi una metrica relativa alla stringa «Error C» nei log. È quindi possibile definire un allarme che reagisca a questa metrica.

Configura le dipendenze concesse in licenza in AWS Transform for mainframe Runtime on container

Questo argomento descrive come configurare dipendenze con licenza aggiuntive che è possibile utilizzare con AWS Transform for mainframe Runtime on container.

Argomenti

- Prerequisiti
- Panoramica di

Prerequisiti

Prima di iniziare, assicurati di completare i seguenti prerequisiti.

- Completa the section called "AWS Prerequisiti di Transform for mainframe Runtime" e. the section called "Onboarding AWS Transform per mainframe Runtime"
- Ottieni le seguenti dipendenze dalla loro fonte.

Oracle database

Fornisci un driver di database Oracle. Ad esempio, ojdbc11-23.3.0.23.09.jar.

Connessione IBM MQ

Fornisci un client IBM MQ. Ad esempio, com.ibm.mq.jakarta.client-9.3.4.1.jar.

Con questa versione di dipendenza, fornisci anche le seguenti dipendenze transitive:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15 a 18-1.76.jar
- bcutil-jdk15 a 18-1.76.jar

File della stampante DDS

Fornisci la libreria di report Jasper (munity.jaspersoft)https://com. com/download-jaspersoft/community-edizione). 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
- commons-beanutils-1.9.4.jar
- commons-collections-3.2.2.jar

Panoramica di

Per installare le dipendenze, completa i passaggi seguenti.

1. Copia una delle dipendenze precedenti come richiesto nella cartella di creazione dell'immagine Docker.
2. Se il tuo database JICS è ospitato su Oracle, inserisci 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 o Amazon EKS.
6. Controllo dei log.

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 `listener-port` i valori corretti per la configurazione.

```
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 configuratoBlusam, puoi aspettarti una risposta vuota quando esegui il comando seguente.

```
curl http://load-balancer-DNS-name:listener-port/bac/api/services/rest/bluesamserver/
serverIsUp
```

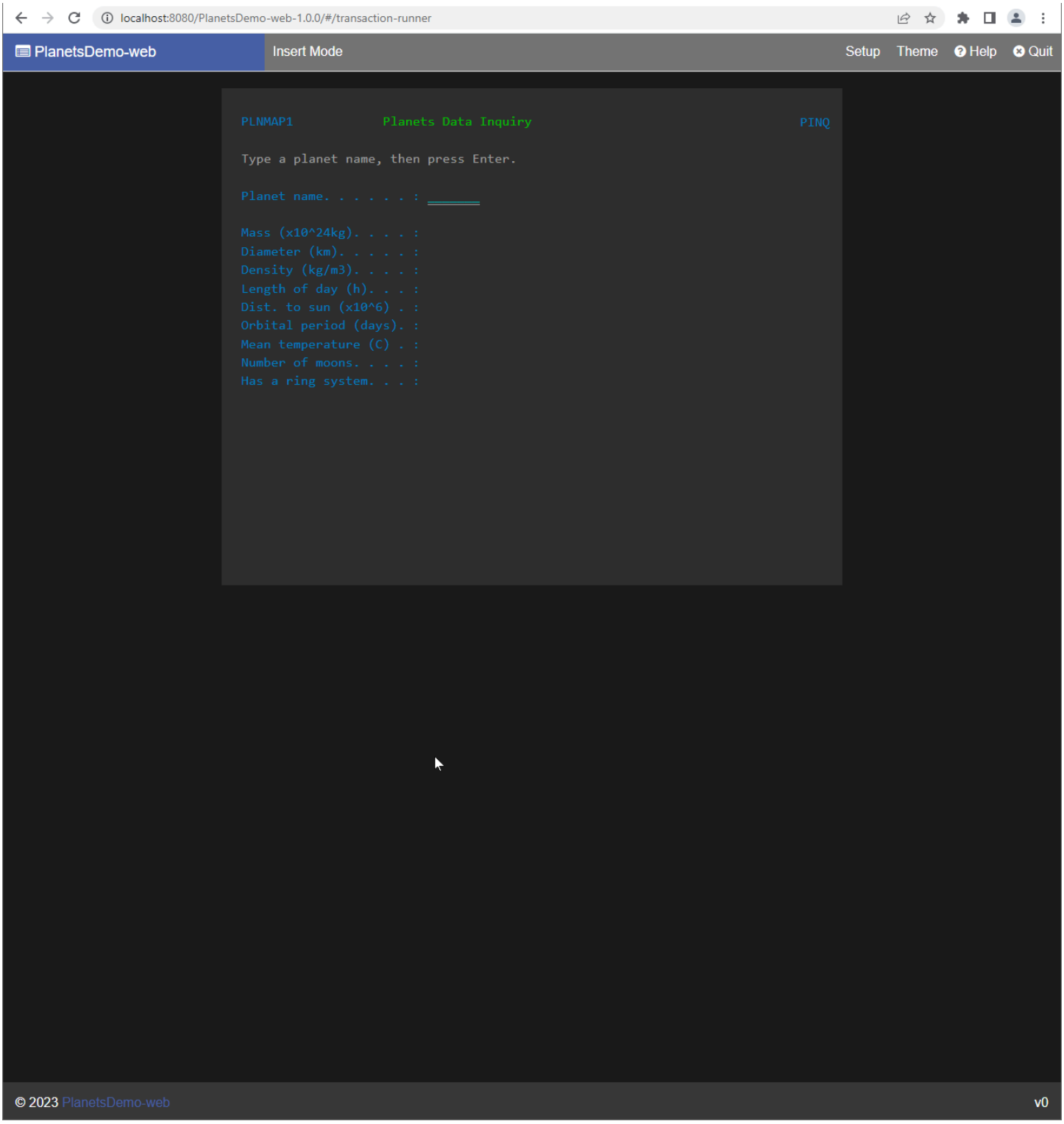
Nota il nome del file binario web (PlanetsDemo-web-1.0.0, se invariato). Per accedere all' PlanetsDemo applicazione, utilizzate un URL nel seguente formato.

```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

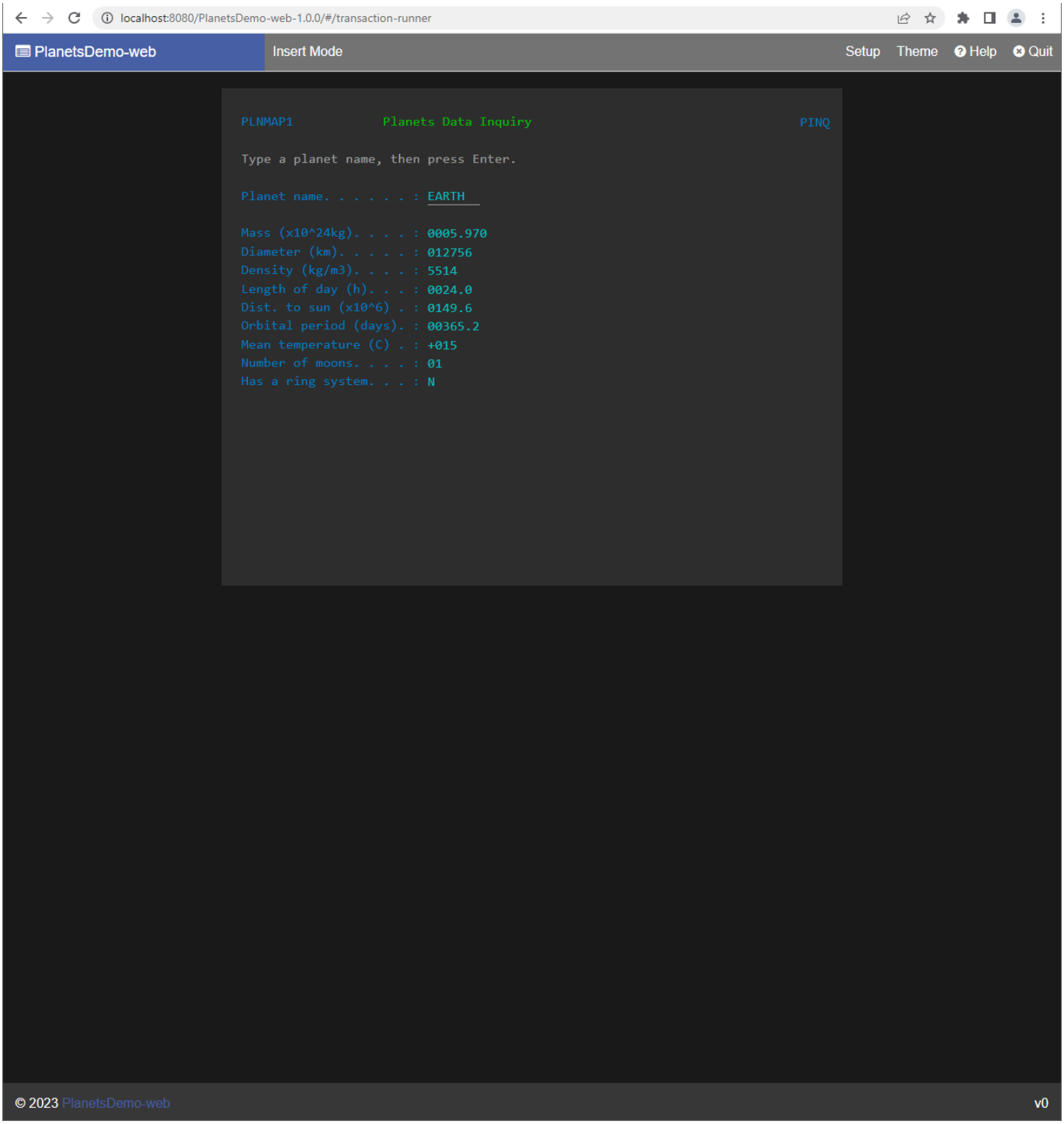
Dopo l'avvio PlanetsDemo dell'applicazione, viene visualizzata la home page.



Immettete PINQ nella casella di testo, quindi premete Invio. Viene visualizzata la pagina di richiesta dati.



Ad esempio, immettete EARTH nel campo del PlanetsDemo nome, quindi premete Invio. Viene visualizzata la pagina relativa al pianeta inserito.


localhost:8080/PlanetsDemo-web-1.0.0/#/transaction-runner
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
© 2023 PlanetsDemo-web
v0

Modifica il codice sorgente con AWS Transform for mainframe Developer IDE

Se utilizzi il motore di runtime AWS Transform for mainframe AWS-managed, puoi utilizzare AWS Transform for mainframe 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 AWS Transform for mainframe Developer tramite Amazon WorkSpaces Applications. Questa sezione descrive come configurare AWS Transform for mainframe Developer on WorkSpaces Applications. Spiega inoltre come utilizzare AWS Transform for mainframe Developer per aggiornare il codice sorgente, utilizzando l'applicazione di esempio. PlanetsDemo

Argomenti

- Tutorial: Configurazione WorkSpaces delle applicazioni per AWS Transform for mainframe Developer IDE
- Tutorial: Usa AWS Transform for mainframe Developer sulle applicazioni WorkSpaces

Tutorial: Configurazione WorkSpaces delle applicazioni per AWS Transform for mainframe Developer IDE

AWS La modernizzazione del mainframe fornisce diversi strumenti tramite Amazon WorkSpaces Applications. WorkSpaces Applications è un servizio di streaming di applicazioni completamente gestito e sicuro che consente di trasmettere applicazioni desktop agli utenti senza riscrivere le applicazioni. WorkSpaces Applications 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 di WorkSpaces Applications to host di 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 del browser nelle WorkSpaces applicazioni, consulta i requisiti di sistema e il supporto delle funzionalità (browser Web) nella Amazon WorkSpaces Applications Administration Guide. In caso di problemi durante l'utilizzo WorkSpaces delle applicazioni, consulta la sezione Risoluzione dei problemi degli utenti AppStream 2.0 nella Amazon WorkSpaces Applications Administration Guide.

Questo documento descrive come configurare AWS Transform for mainframe Developer IDE su una flotta di WorkSpaces applicazioni.

Argomenti

- Prerequisiti
- Fase 1: creazione di un bucket Amazon S3
- Passaggio 2: allega una policy al bucket S3
- Fase 3: caricare i file nel bucket Amazon S3
- Fase 4: Scarica i modelli CloudFormation
- Fase 5: Creare la flotta con CloudFormation
- Fase 6: Accedere a un'istanza
- Eseguire la pulizia delle risorse

Prerequisiti

Per gli utenti alle prime armi, procedi nel seguente modo:

1. Passa alla console AppStream 2.0 a https://console.aws.amazon.com/appstream2/casa.
2. Seleziona Inizia.
3. Scegliere Skip (Ignora).

> **Important**
>
> Amazon WorkSpaces Applications utilizza i ruoli IAM per gestire le risorse delle WorkSpaces applicazioni e li AWS creerà quando lo farai.

Quindi, scarica il file di archivio che contiene gli elementi necessari per configurare AWS Transform for mainframe Developer IDE in Applications. WorkSpaces

> **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. Per ulteriori informazioni sull'avvio e la connessione a un'istanza Amazon EC2, consulta Get started with Amazon EC2.

Fase 1: creazione di un bucket Amazon S3

Crea un bucket Amazon S3 nella Regione AWS stessa flotta di WorkSpaces applicazioni che creerai. Questo bucket conterrà gli artefatti necessari per completare questo tutorial. Per ulteriori informazioni sui bucket, consulta Creazione di un bucket.

Passaggio 2: allega una policy al bucket S3

Allega la seguente policy al bucket che crei per questo tutorial. Per ulteriori informazioni su come allegare una policy al bucket S3, consulta Aggiungere una policy bucket.

Assicurati di sostituirlo `amzn-s3-demo-bucket` con il nome effettivo del bucket che hai creato.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "AllowAppStream2.0ToRetrieveObjects",
        "Effect": "Allow",
        "Principal": {
            "Service": "appstream.amazonaws.com"
        },
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }]
}
```

Fase 3: caricare 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 CloudFormation

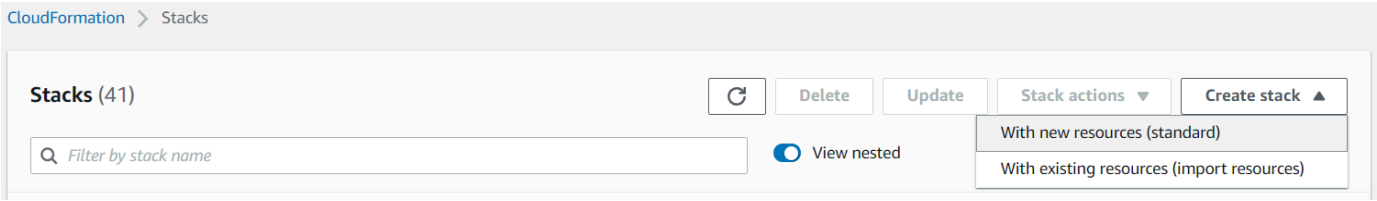
Scarica i seguenti CloudFormation modelli. Questi modelli sono necessari per creare e popolare il parco di WorkSpaces applicazioni.

- 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

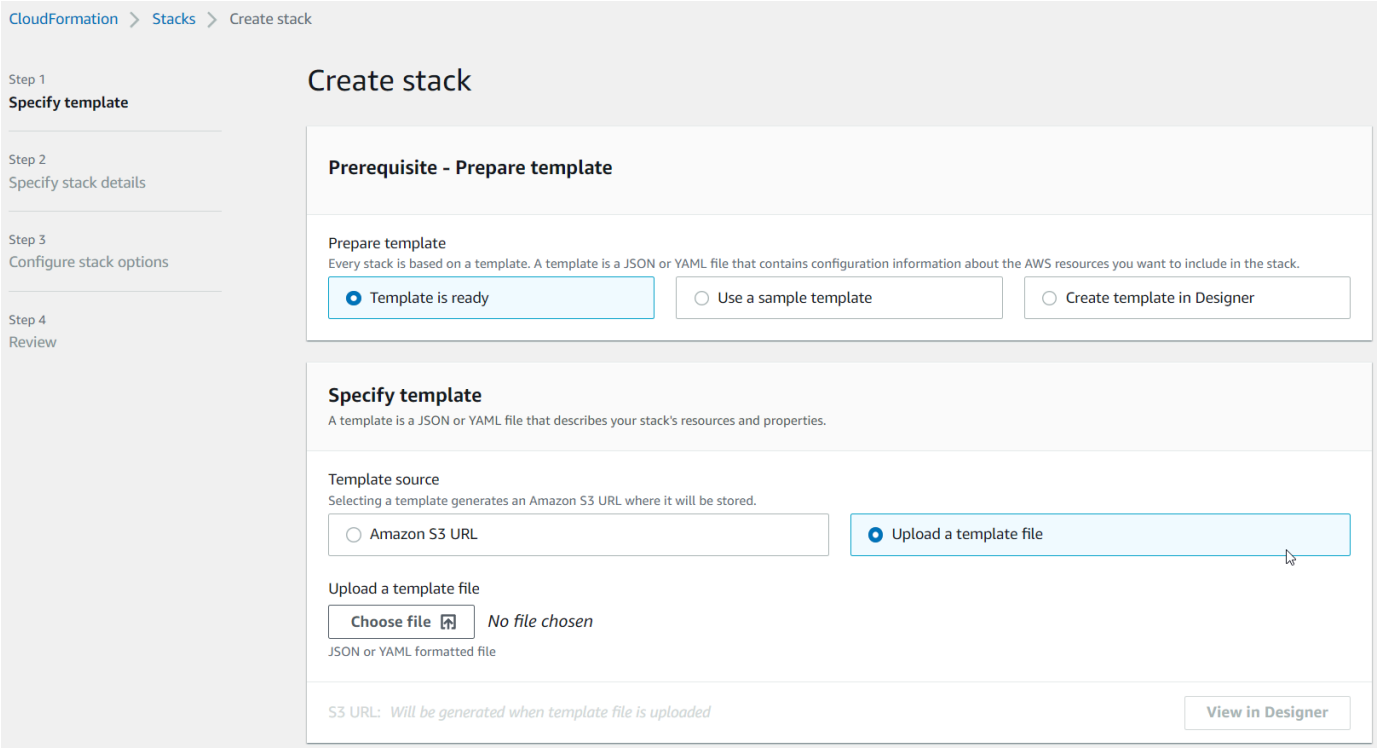
Fase 5: Creare la flotta con CloudFormation

In questo passaggio, utilizzerai il `cfn-m2-appstream-elastic-fleet-linux.yaml` CloudFormation modello per creare una flotta e uno stack di WorkSpaces applicazioni per ospitare l'IDE AWS Transform for mainframe Developer. Dopo aver creato la flotta e lo stack, eseguirai gli altri CloudFormation modelli scaricati nel passaggio precedente per installare l'IDE per sviluppatori e gli altri strumenti richiesti.

1. Vai alla console CloudFormation di AWS gestione e scegli Stacks.
2. In Stacks, scegli Crea stack e Con nuove risorse (standard):



3. In Crea stack, scegli Scegli un modello esistente e Carica un file modello:



4. Scegli il file e vai al file`cfn-m2-appstream-elastic-fleet-linux.yaml`. Scegli Next (Successivo).
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 Next (Successivo).
7. Naviga verso il basso nella pagina e scegli Riconosco che CloudFormation potrebbe creare risorse IAM con nomi personalizzati. .
8. Scegli Next (Successivo).
9. Controlla i dettagli e scegli Invia.
10. Dopo aver creato la flotta, crea CloudFormation pile con tutti 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.

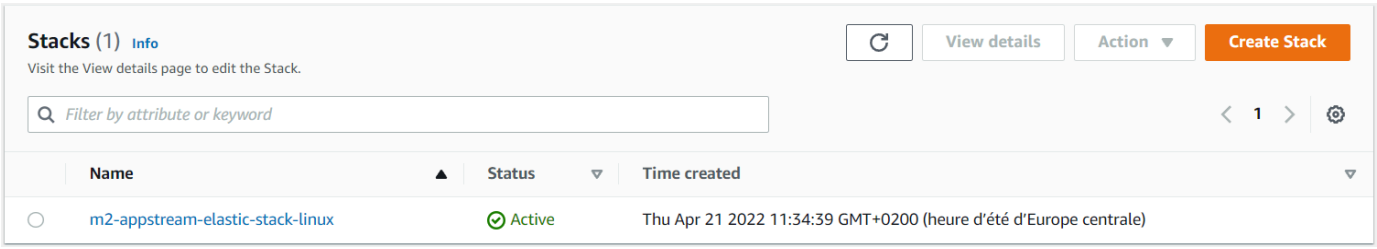
> **Note**
>
> I modelli scaricati prevedono di trovare risorse in un bucket S3 con una struttura di cartelle denominata. `appstream/bluage/developer-ide/` Il bucket deve appartenere alla Regione AWS stessa flotta che hai creato.

> **Important**
>
> Esegui tutti gli CloudFormation script scaricati nel passaggio 4 per configurare correttamente l'applicazione.

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. Vai su WorkSpaces Applicazioni in Console di gestione AWS e scegli lo stack creato in precedenza:



2. Nella pagina dei dettagli dello stack, scegli lo stack, quindi scegli Associate fleet.
3. Nel prompt, scegli la flotta che hai creato e avviato in precedenza.
4. Selezionare Associate (Associa).
5. Scegli lo stack associato e dal menu Azioni scegli 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.

Eseguire la pulizia delle risorse

Per la procedura di pulizia dello stack e delle flotte creati, consulta Create an WorkSpaces Applications Fleet and Stack.

Dopo aver eliminato gli oggetti WorkSpaces Applications, tu o l'amministratore dell'account potete anche pulire i bucket S3 per le impostazioni delle applicazioni 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 di WorkSpaces applicazioni.

Non puoi utilizzare la console WorkSpaces Applicazioni per eliminare utenti. È invece necessario utilizzare l'API del servizio con AWS CLI. Per ulteriori informazioni, consulta User Pool Administration nella Amazon WorkSpaces Applications Administration Guide.

Tutorial: Usa AWS Transform for mainframe Developer sulle applicazioni WorkSpaces

Questo tutorial mostra come accedere a AWS Transform for mainframe Developer on WorkSpaces Applications e utilizzarlo con un'applicazione di esempio in modo da poterne provare le funzionalità. Al termine di questo tutorial, puoi seguire gli stessi passaggi con le tue 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
- Passaggio 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
- Eseguire la 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. Gestione dei segreti AWS Per ulteriori informazioni, consulta la sezione Gestione delle password con Amazon RDS e Gestione dei segreti AWS nella Guida per l'utente di Amazon RDS.
6. Assicurati che il VPC sia lo stesso che usi per l'istanza WorkSpaces Applications. 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, si accede all'ambiente di sviluppo AWS Transform for mainframe su WorkSpaces Applications.

1. Contatta l'amministratore per sapere come accedere correttamente all'istanza di WorkSpaces Applications. Per informazioni generali sui possibili client e configurazioni, consulta WorkSpaces Applications Access Methods and Clients nella Amazon WorkSpaces Applications Administration Guide. Valuta la possibilità di utilizzare il client nativo per un'esperienza ottimale.
2. In WorkSpaces Applicazioni scegli Desktop.

Fase 3: Configurare il runtime

In questo passaggio, si configura il runtime AWS Transform for mainframe. 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.

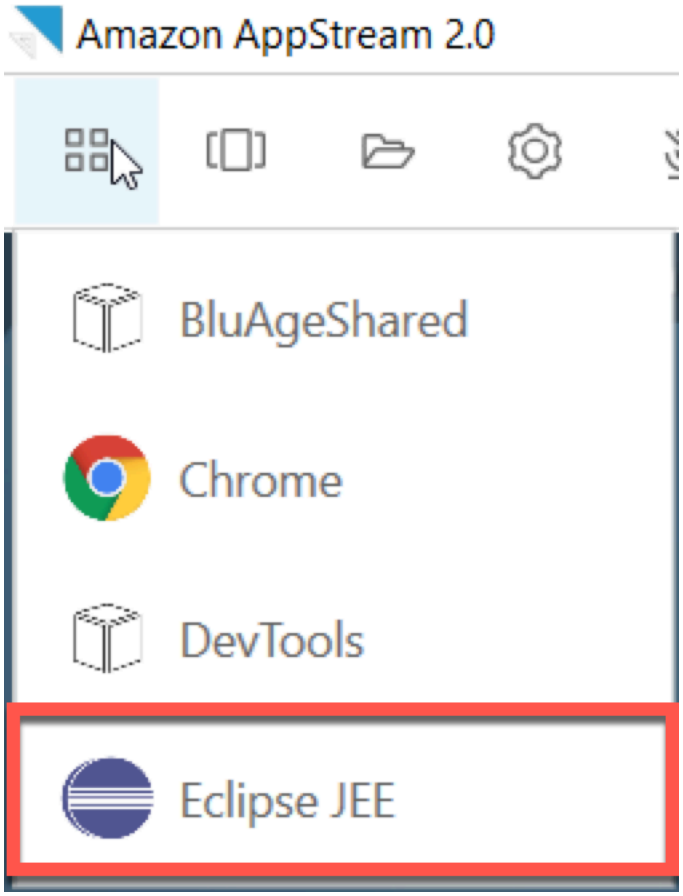
1. Scegli Applicazioni, dalla barra dei menu, quindi scegli Terminale.
2. Immetti il comando seguente:

```
~/_install-velocity-runtime.sh
```

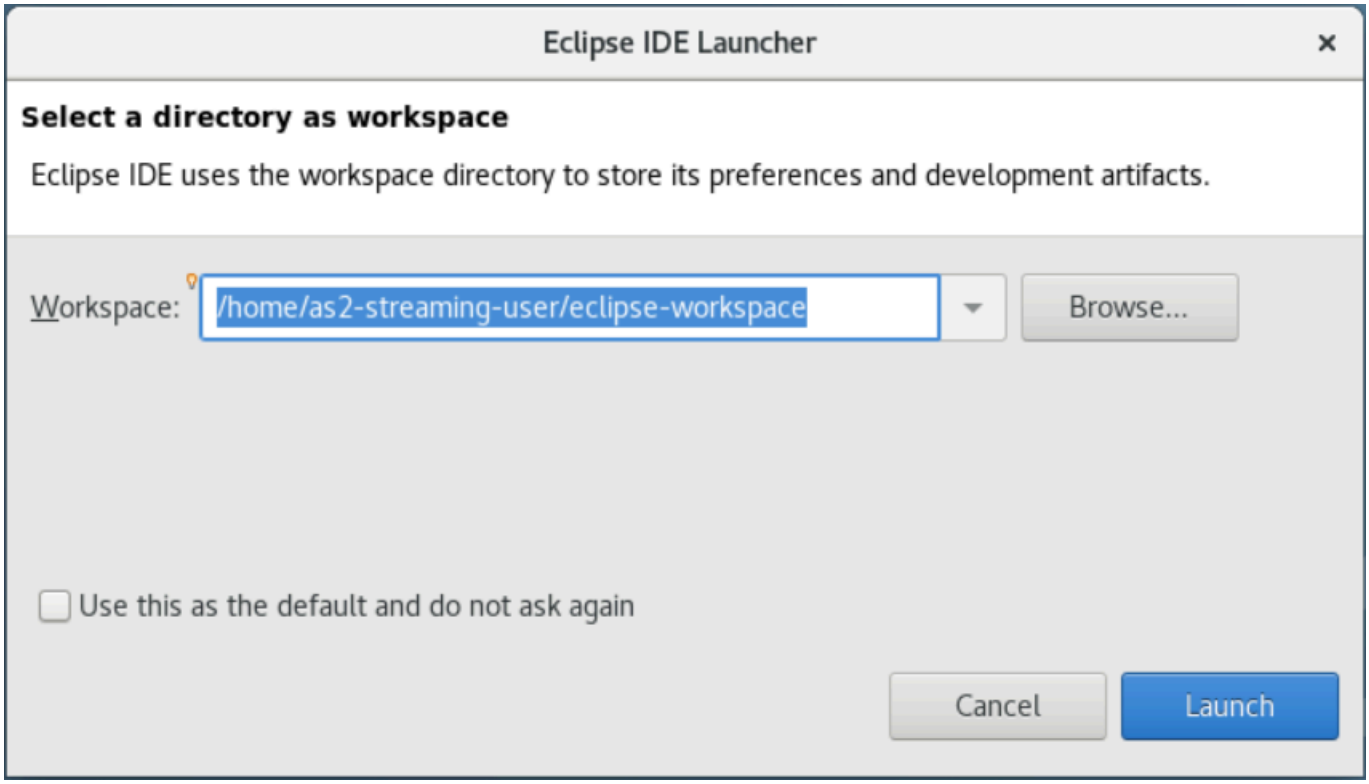
Passaggio 4: Avvia l'IDE di Eclipse

In questo passaggio, avvii l'IDE di Eclipse e scegli una posizione in cui desideri creare uno spazio di lavoro.

1. In WorkSpaces Applicazioni, 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 Transform for mainframe installato.

   Se non siete sicuri della versione installata, eseguite il seguente comando in un terminale:

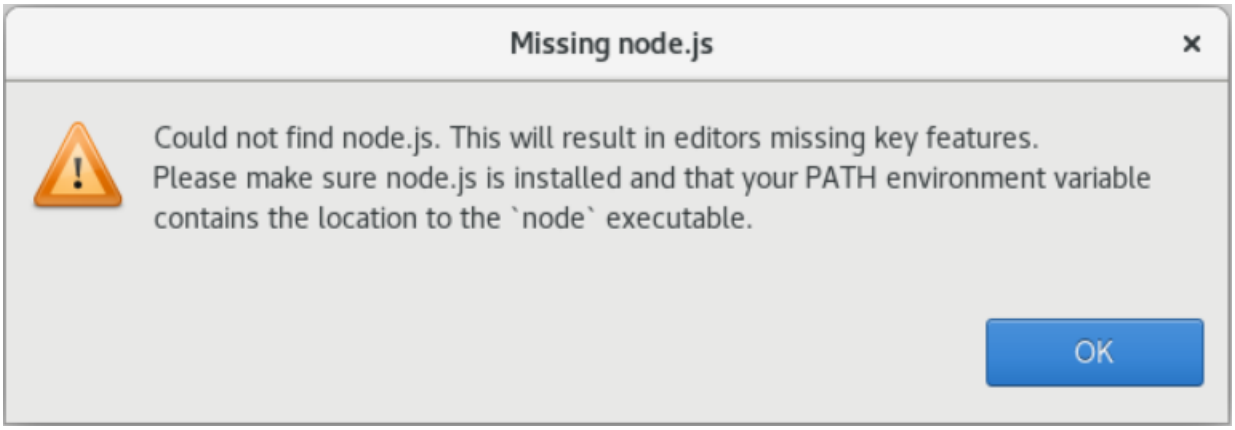
   ```
   cat ~/runtime-version.txt
   ```

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. Scegli Next (Successivo).
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:

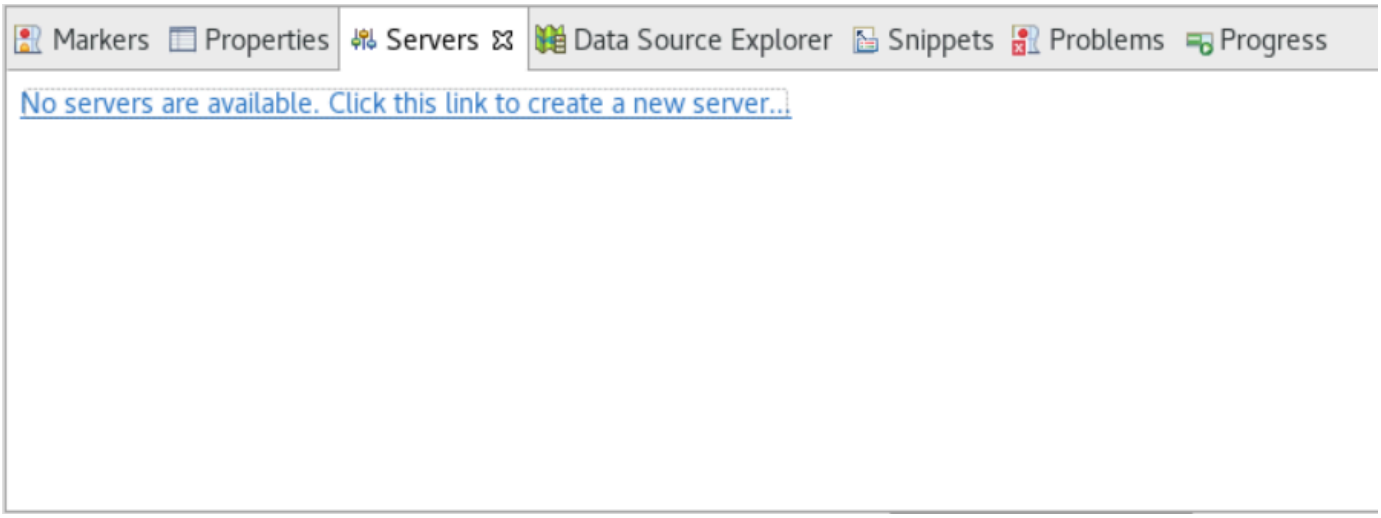


   Attendi fino alla fine della compilazione. Puoi seguire la build nella vista Progress.
7. In Eclipse, seleziona il progetto e scegli Esegui come. Quindi scegli Maven install. Una volta completata l'installazione di Maven, crea il file seguente. `war` `PlanetsDemoPom/PlanetsDemo-web/target/PlanetsDemo-web-1.0.0.war`

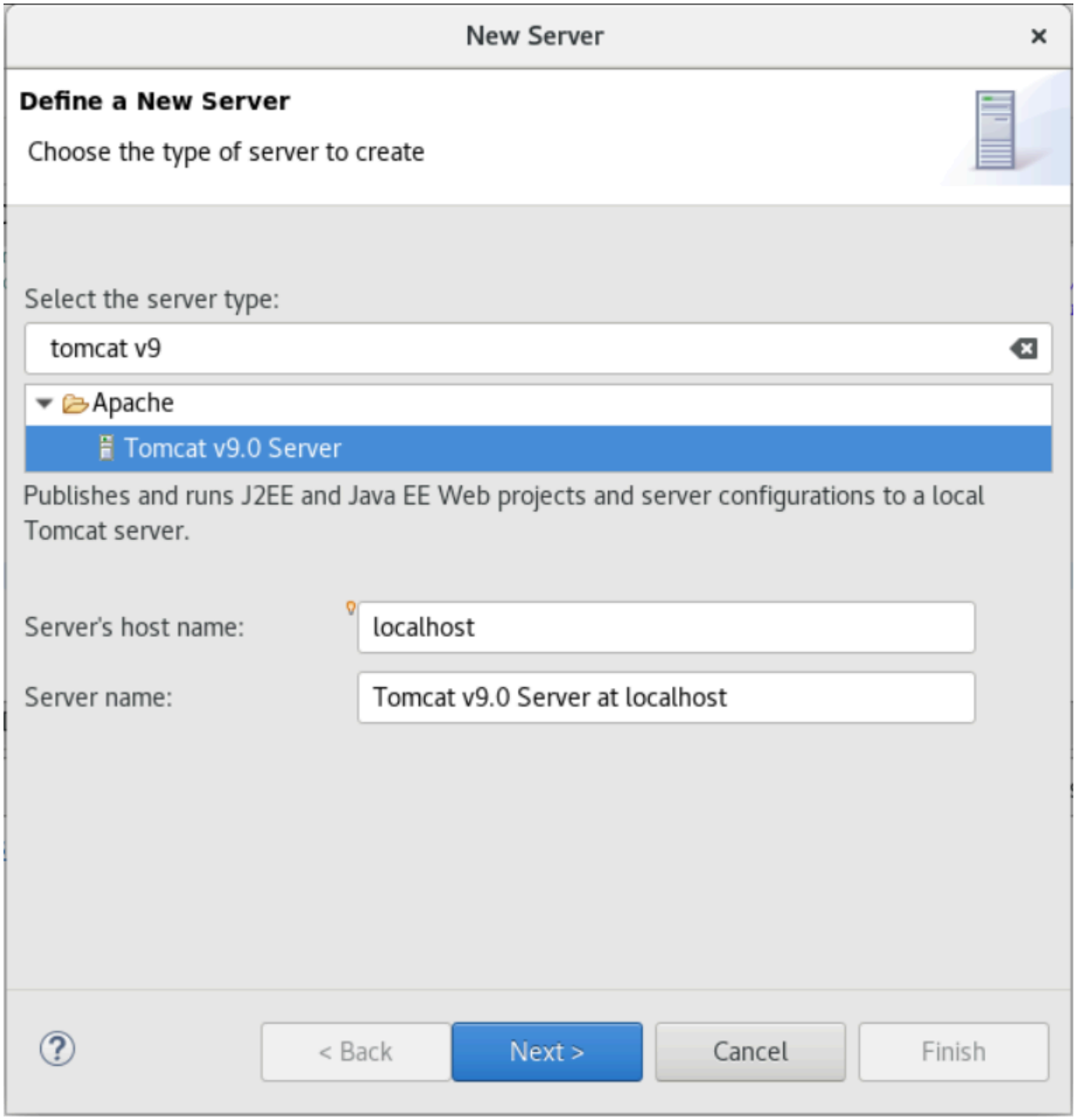
Passaggio 6: Configurare un server Tomcat

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:

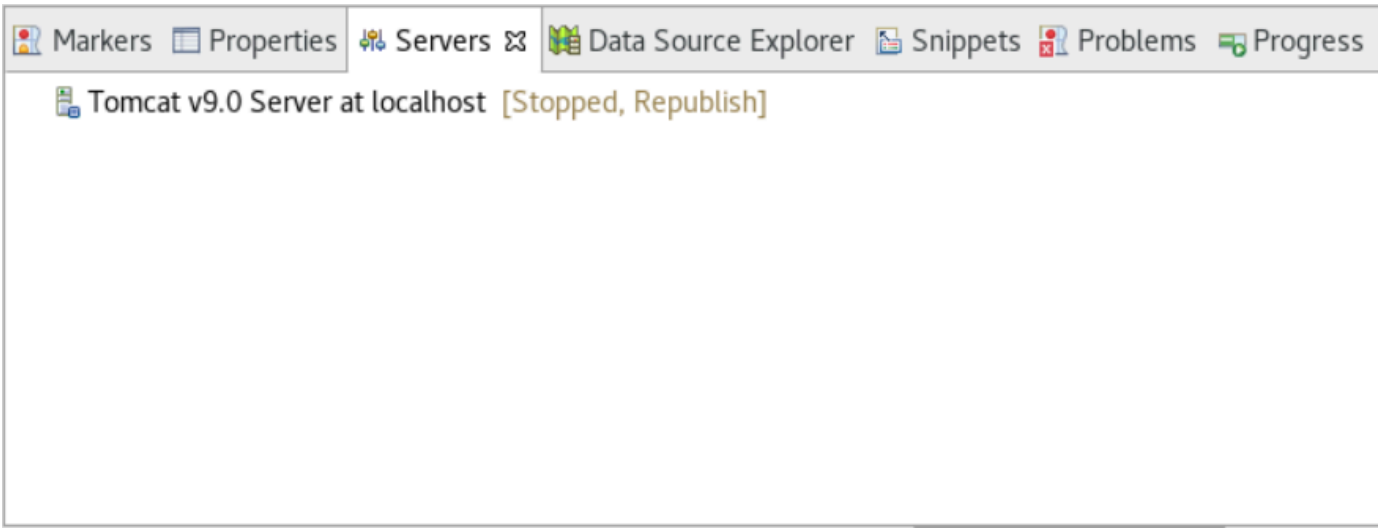


2. Scegli Non ci sono server disponibili. Fai clic su questo link per creare un nuovo server... . Viene visualizzata la procedura guidata New Server. Nel campo Seleziona il tipo di server della procedura guidata, inserisci tomcat v9 e scegli Tomcat v9.0 Server. Quindi scegli Successivo.



3. Scegli Sfoglia e scegli la cartella tomcat nella radice della cartella Home. Lascia JRE al valore predefinito e scegli Fine.

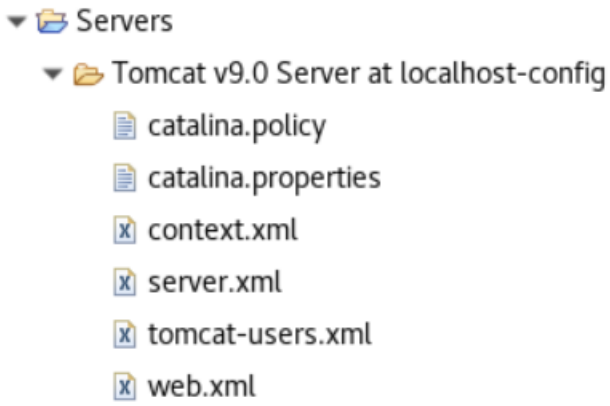
   Un progetto Servers viene creato nell'area di lavoro e un server Tomcat v9.0 è ora disponibile nella vista Server. Qui verrà distribuita e avviata l'applicazione compilata:



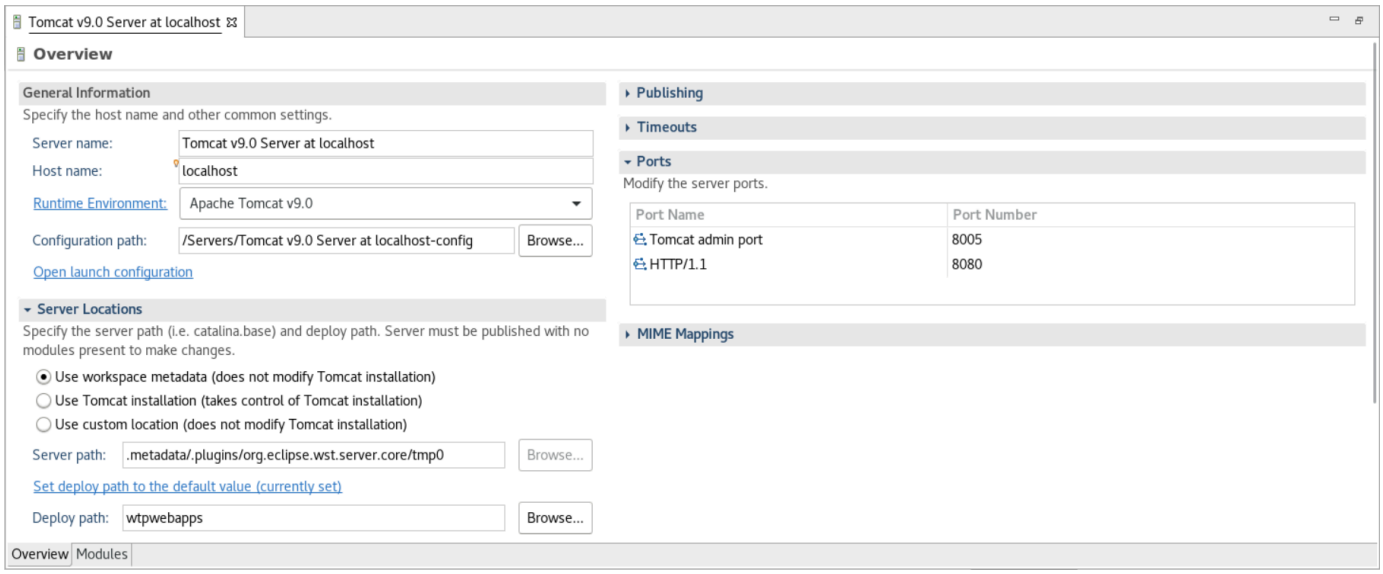
Passaggio 7: Esegui la distribuzione su Tomcat

In questo passaggio, distribuisci l'applicazione demo Planets sul server Tomcat in modo da poter eseguire l'applicazione.

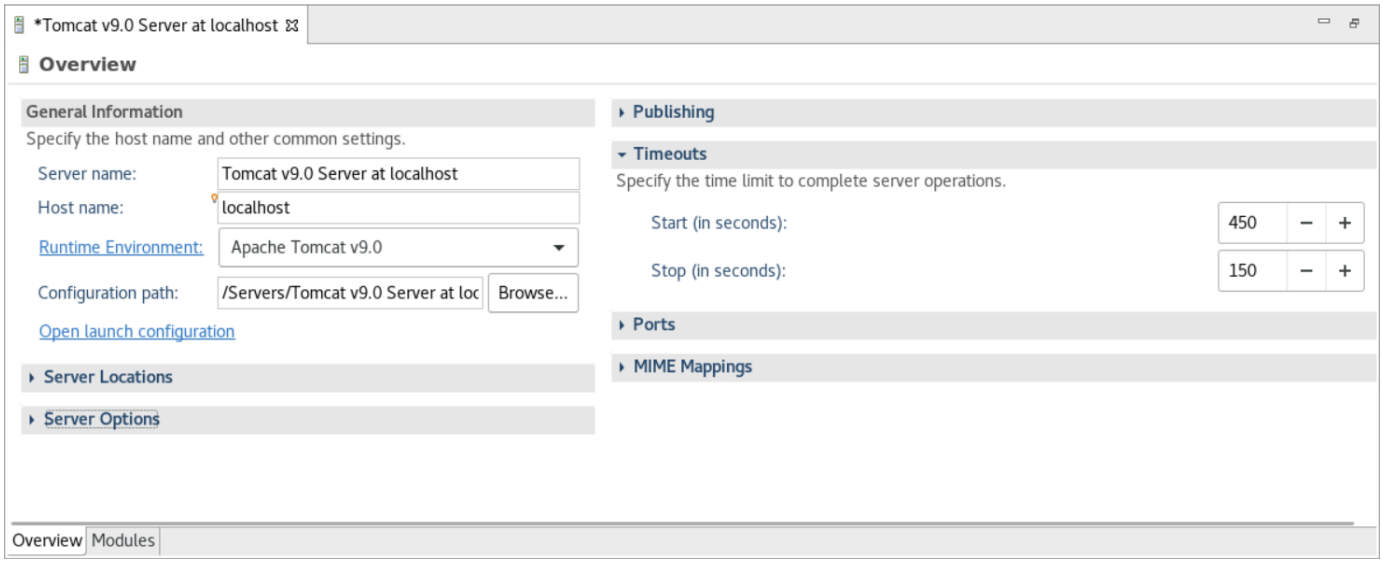
1. Seleziona il `PlanetsDemo-web` file e scegli Esegui come > Maven install. Seleziona `PlanetsDemo-web` di nuovo e scegli Aggiorna per assicurarti che il frontend compilato con npm sia correttamente compilato in .war e notato da Eclipse.
2. Carica PlanetsDemo-runtime.zip sull'istanza e decomprimi il 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:



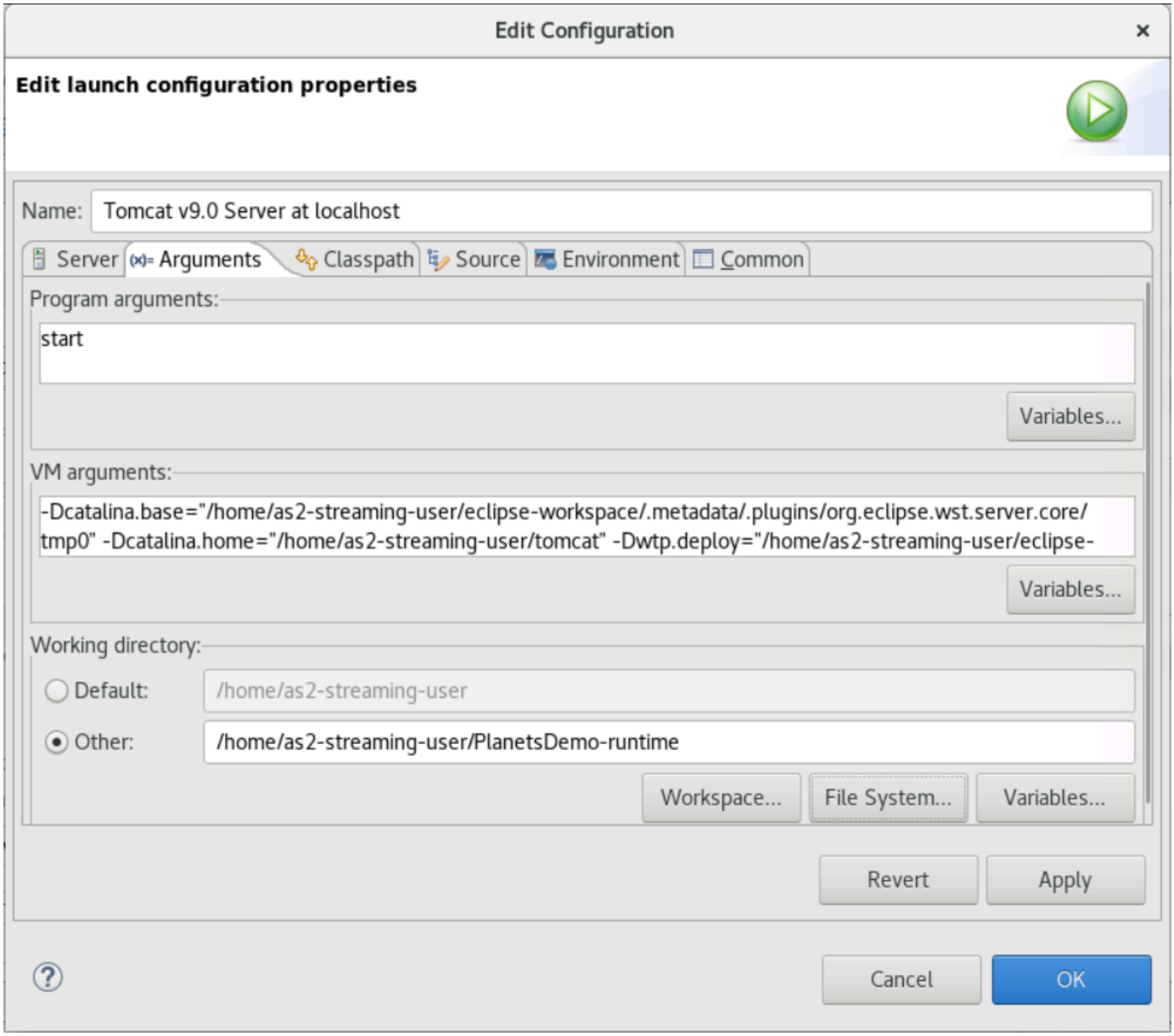
4. Aprire la voce relativa al `tomcat v9.0` server nella vista Server. Viene visualizzato l'editor delle proprietà del server:



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.

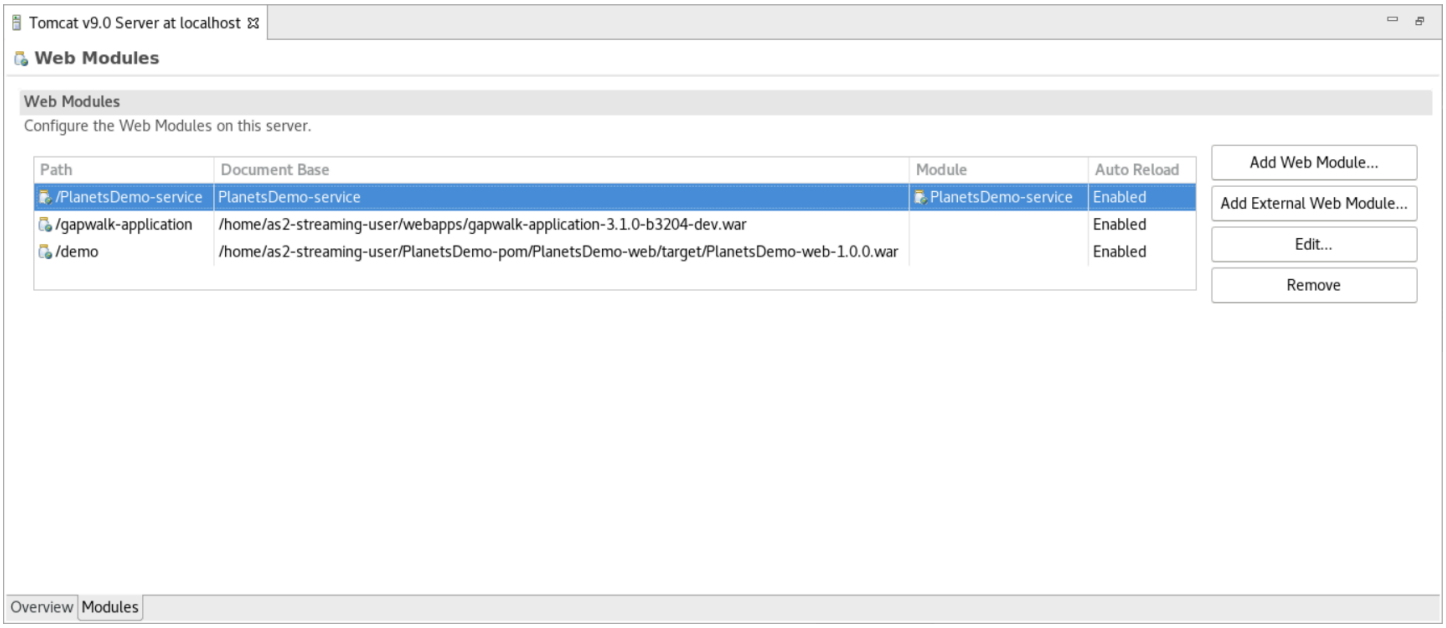


7. Scegliete la scheda Moduli dell'editor delle proprietà del server e apportate le seguenti modifiche:

   - 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, inserisci `/demo`
  - Seleziona OK
  - Salva le modifiche all'editor (Ctrl + S).

Il contenuto dell'editor dovrebbe ora essere simile al seguente.



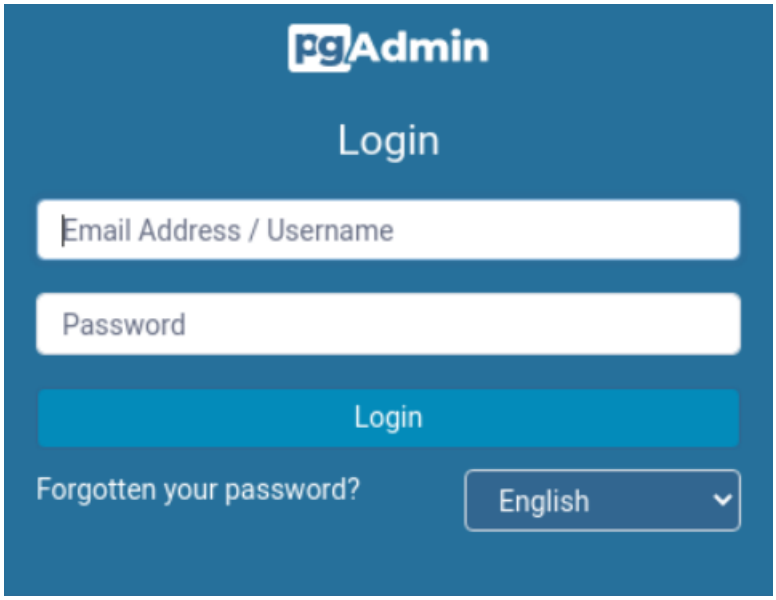
Fase 8: Creare il database JICS

In questo passaggio, ti connetti al database in Fase 1: crea un database cui hai creato.

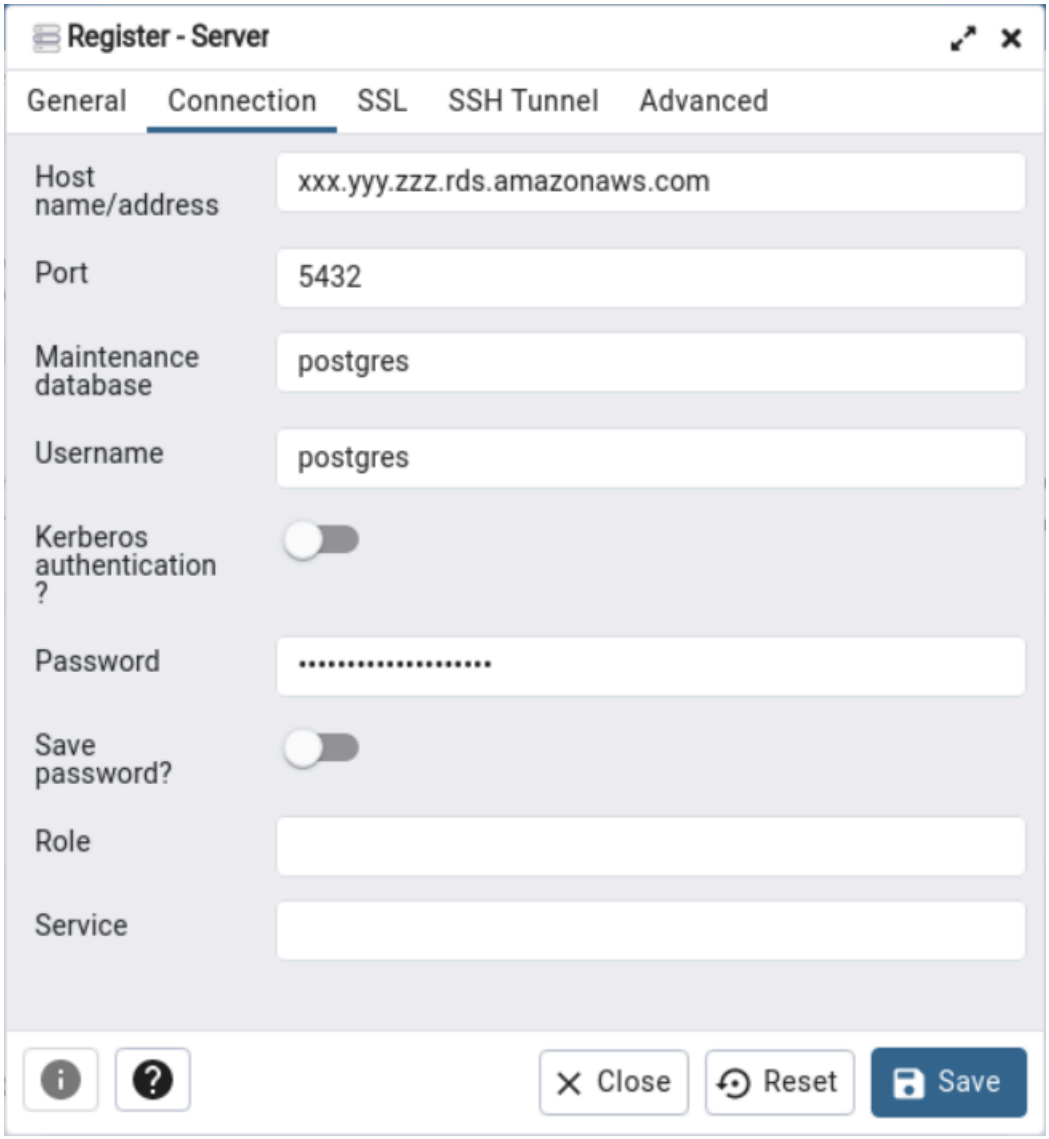
1. Dall'istanza WorkSpaces Applications, esegui il seguente comando in un terminale per avviare`pgAdmin`:

   ```
   ./pgadmin-start.sh
   ```

2. Scegli un indirizzo e-mail 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.

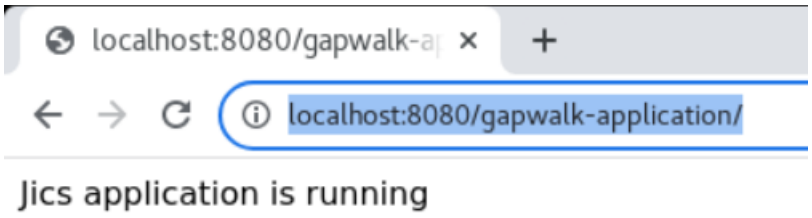


4. Quando ti connetti al server del database, usa Oggetto > Crea > Database e crea un nuovo database denominato jics.
5. Modifica le informazioni di connessione al database utilizzate dall'app demo. Queste informazioni sono definite in`PlanetsDemo-runtime/config/application-main.yml`. Cerca la `jicsDs` 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.

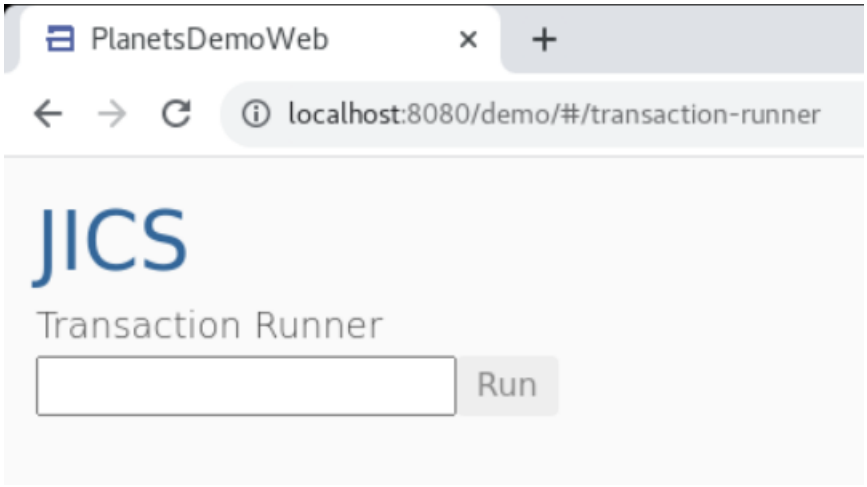
Passaggio 9: avviare e testare l'applicazione

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.

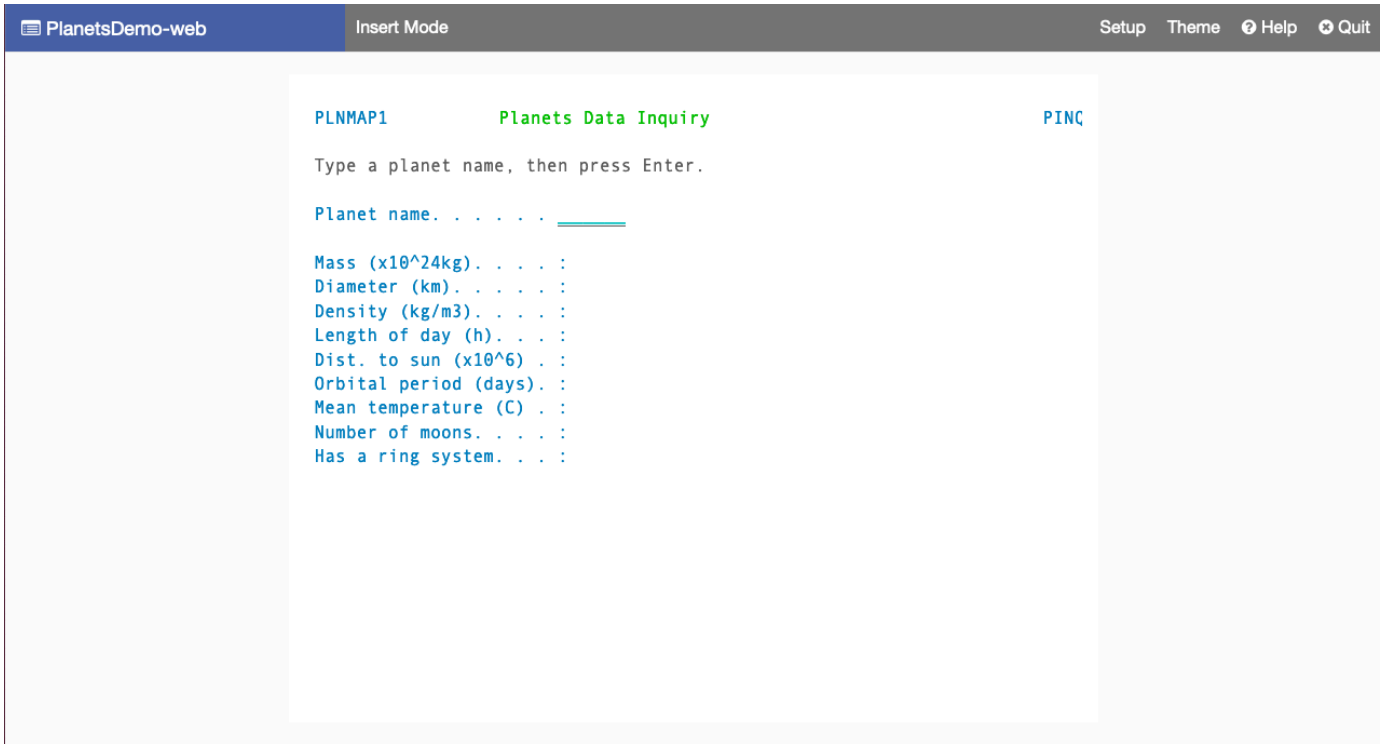


3. Accedi al frontend dell'applicazione distribuita da Google Chrome all'indirizzo http://localhost:8080/demo. Dovrebbe apparire la seguente pagina Transaction Launcher.

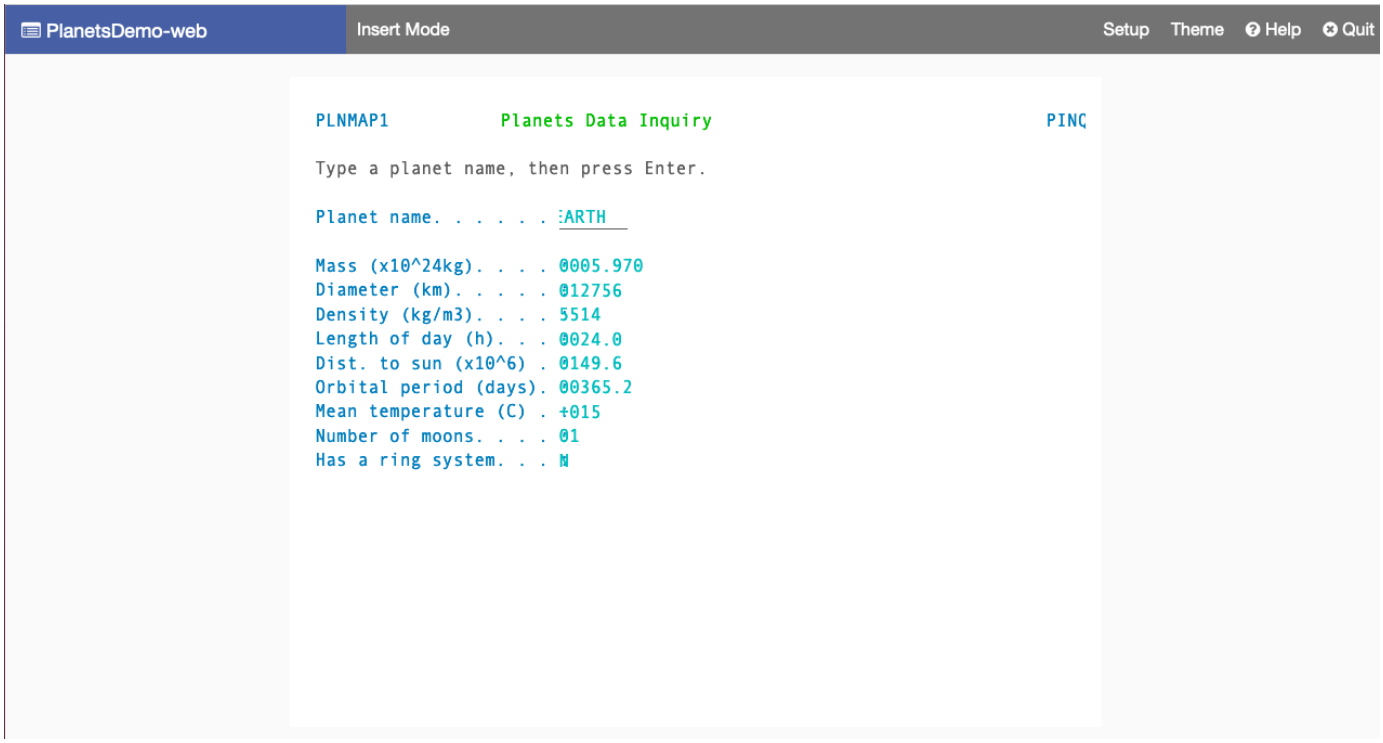


4. Per avviare la transazione dell'applicazione, inserisci `PINQ` nel campo di immissione e scegli Esegui (o premi Invio).

   Dovrebbe apparire la schermata dell'app demo.



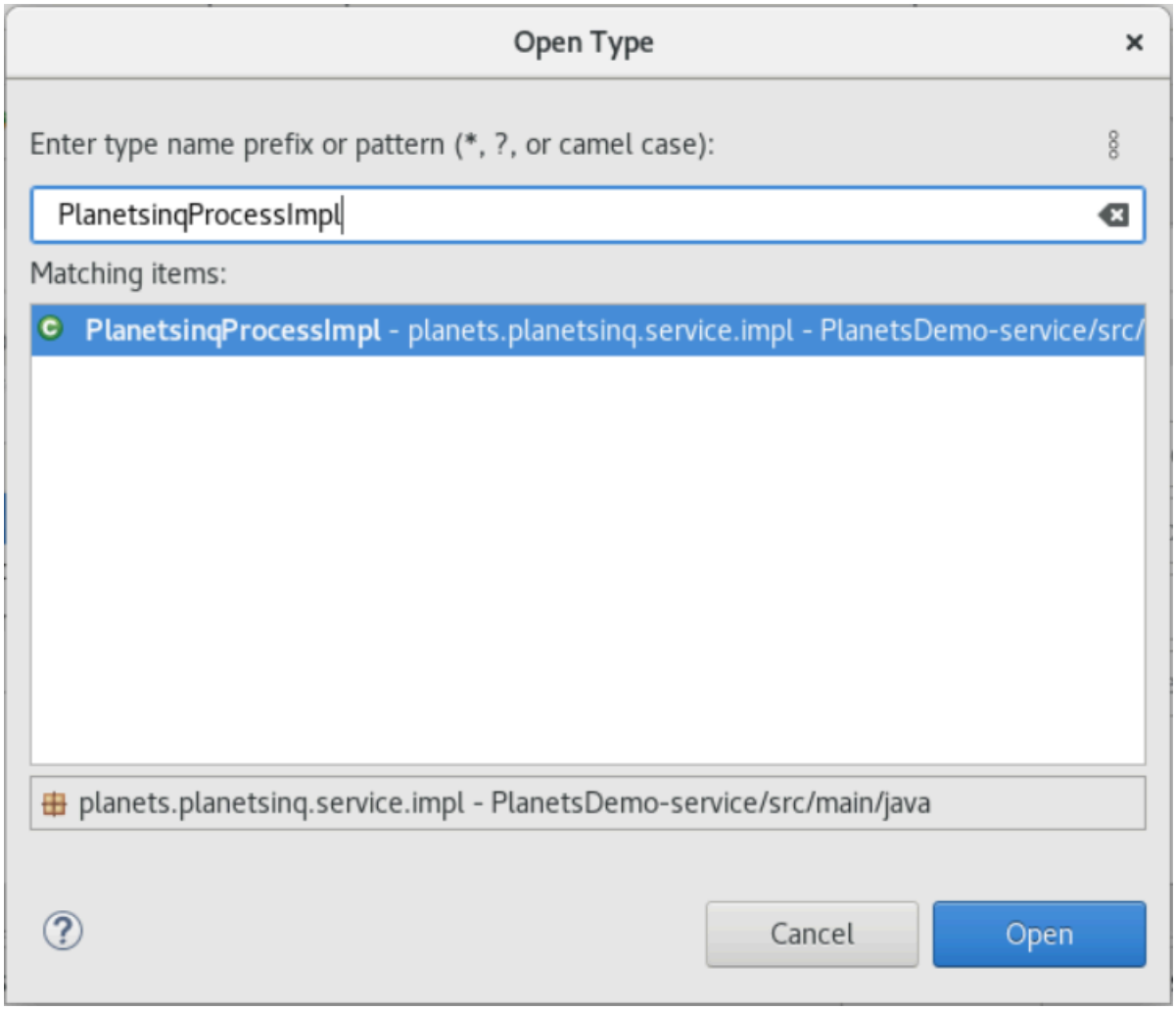
5. Digita il nome di un pianeta nel campo corrispondente e premi Invio.



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.

Eseguire la 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 creatoFase 1: crea un database. Per ulteriori informazioni, consulta la sezione relativa all'eliminazione di un'istanza database.

AWS Domande frequenti su Transform for mainframe

Ambito generale

1. Qual è lo scopo principale della funzionalità di AWS refactoring di Transform for mainframe?

   La funzionalità di refactoring rifattorizza il codice monolitico legacy in java utilizzando applicazioni distribuite contemporanee che utilizzano linguaggi e framework moderni, seguendo un modello di refactoring automatizzato. Questo modello prevede l'analisi automatica del codice legacy, la comprensione della sua funzionalità e la conversione in codice moderno equivalente, preservando al contempo la logica di business. Il processo include la modernizzazione non solo del codice, ma anche dell'intero stack applicativo, delle dipendenze e dell'infrastruttura utilizzando strumenti e processi automatizzati. La soluzione mira ad accelerare la modernizzazione mantenendo l'equivalenza funzionale e le prestazioni. Ciò include la trasformazione sia del codice dell'applicazione che dei database e degli archivi di dati associati, implementando al contempo le migliori pratiche e i modelli di progettazione del cloud.
2. Quali applicazioni mainframe sono supportate da AWS Transform for mainframe?

   AWS Transform for mainframe attualmente supporta la modernizzazione delle applicazioni z/OS mainframe IBM scritte in COBOL, PL/I, JCL (Job Control Language) e si basano sul gestore delle transazioni CICS (Customer Information Control System), schermate BMS (Basic Mapping Support), schermi IMS MFS, database IMS, file flat, file di dati GDG (Generation data groups) e VSAM (Virtual Storage Access Method). DB2
3. Quali linguaggi mainframe può modernizzare AWS Transform for mainframe?

   AWS Transform for mainframe trasforma COBOL e il PL/I codice in Java, in Groovy, JCLs gli schermi (BMS o MFS) in HTML (con Sass) e JavaScript (applicazioni Angular: per ora React

non è supportato), consentendo la modernizzazione delle applicazioni mainframe legacy verso architetture native per il cloud. Queste tecnologie sono state scelte per la loro adozione diffusa, il robusto ecosistema e le funzionalità native del cloud. Angular offre un livello di interfaccia utente moderno e reattivo che sostituisce le interfacce a schermo verde precedenti. Consente la creazione di applicazioni Web dinamiche e di facile utilizzo a cui è possibile accedere su diversi dispositivi e piattaforme. La sua architettura basata su componenti supporta uno sviluppo front-end gestibile e scalabile. La trasformazione si traduce in applicazioni distribuite che seguono modelli architettonici moderni e migliori pratiche.

4. In che modo AWS Transform for mainframe bilancia i vincoli esistenti con i vantaggi del cloud?

   AWS Transform for mainframe raggiunge l'equilibrio preservando la logica e le funzionalità aziendali critiche e introducendo funzionalità native del cloud. Garantisce che le applicazioni modernizzate mantengano la logica di business legacy necessaria, sfruttando al contempo la scalabilità, l'agilità e le moderne pratiche operative del cloud. Questo approccio aiuta le organizzazioni a mantenere la continuità aziendale ottenendo al contempo i vantaggi dell'infrastruttura cloud.

5. Che ruolo gioca l'architettura orientata ai servizi nell'applicazione modernizzata?

   L'architettura orientata ai servizi svolge un ruolo fondamentale nella suddivisione delle applicazioni monolitiche in componenti modulari più gestibili. AWS Transform for mainframe crea applicazioni orientate ai servizi e agli oggetti che facilitano una migliore manutenibilità e scalabilità. Questo approccio architetturale consente alle organizzazioni di raggiungere una maggiore efficienza aziendale e prepararsi per la potenziale adozione futura dei microservizi.

6. Quali aspetti dello stack di applicazioni sono inclusi nel processo di refactoring?

   Il processo di refactoring include lo stack software completo: codice dell'applicazione, dipendenze, database e infrastruttura (ad esempio opzioni per la memorizzazione nella cache, supporto per la messaggistica, ecc.). Copre la trasformazione dei linguaggi di programmazione, dei sistemi di database, dei file di dati e dei componenti dell'infrastruttura associati. Questo approccio completo garantisce che tutti gli aspetti dell'applicazione siano modernizzati in modo coerente, con il risultato di uno stack di applicazioni moderno completamente trasformato.

7. Il processo di modernizzazione di AWS Transform for mainframe elimina la necessità di test o controlli di qualità sull'applicazione Java modernizzata?

   No, il processo di modernizzazione di AWS Transform for mainframe non elimina la necessità di test o controlli di qualità sull'applicazione Java modernizzata.

8. Cosa significa AWS Transform for mainframe JAC?

JAC sta per JICS Administration Console

9. Come posso accedere agli strumenti AWS Transform for mainframe?

   AWS Gli strumenti Transform for mainframe sono accessibili tramite la console AWS tramite AWS Mainframe Modernization (M2) Refactor, con accesso alle funzionalità in base al livello di accreditamento. Inizia con il Transformation Center per valutare il refactoring automatico in Java del tuo codice sorgente. Per indicazioni dettagliate, consulta la documentazione di AWS AWS Transform for mainframe refactor. Dopo la modernizzazione, puoi distribuire le applicazioni utilizzando il runtime. Per ulteriori informazioni, consulta la documentazione sulla modernizzazione del AWS mainframe.

10. Come dimensionare (carico di lavoro e tempistica) un progetto?

    Per ulteriori informazioni, consulta AWS Transform for mainframe Refactor Estimates o collabora con il tuo Account Manager.

11. Esistono requisiti specifici per mantenere Java AWS Transform per soluzioni migrate mainframe?

    No, non ci sono requisiti specifici per la manutenzione di Java AWS Transform per soluzioni mainframe migrate.

12. Quali sono le specifiche tecniche e la compatibilità del codice generato da AWS Transform for mainframe?

    AWS Il codice generato da Transform for mainframe è progettato con caratteristiche tecniche specifiche e un'ampia compatibilità. Sebbene non supporti JPA, utilizza l'esecuzione SQL diretta con query esternalizzate. Il codice si basa su librerie specifiche del runtime per l'equivalenza funzionale, la generazione di servizi Web e le implementazioni MQ. Il codice generato può essere importato in qualsiasi IDE Java per lo sviluppo, il test, la creazione e la distribuzione, sebbene le librerie richieste debbano essere importate di conseguenza. Sebbene Maven sia integrato di default con il servizio AWS Mainframe Modernization per i processi di compilazione, è possibile utilizzare strumenti alternativi come Gradle modificando il formato di imballaggio dopo la trasformazione. La piattaforma offre flessibilità in termini di strumenti di sviluppo e controllo del codice sorgente, con corsi di formazione disponibili per i team di sviluppo che gestiscono il codice. Per ulteriori informazioni, consulta AWS Transform for mainframe Runtime High Level Architecture.

AWS Transform for mainframe Runtime

1. Dove posso trovare informazioni su AWS Transform for mainframe Runtime?

Fai riferimento al Transform for mainframeruntime della documentazione di Set up AWS Runtime che descrive in dettaglio il processo di configurazione, l'onboarding, il recupero degli artefatti, la distribuzione, ecc.

2. Le dipendenze JAR di AWS Transform for mainframe vengono caricate nel repository Maven del client per lo sviluppo locale?

   Le librerie possono essere importate in EC2 utilizzando un'AMI che può essere utilizzata per configurare l'ambiente di sviluppo, test e produzione. La formazione e l'abilitazione verranno fornite al team per il codice dell'applicazione generato maintain/enhance . Per ulteriori informazioni, consulta AWS Transform for mainframe Runtime, architettura di alto livello.

3. A cosa si riferisce il termine «Gapwalk» nei jar distribuiti di AWS Transform for mainframe Runtime?

   Per informazioni su Gapwalk, vedete AWS Transform for mainframe Runtime artifacts.

4. Come richiedere l'accesso a Transform for mainframe Runtime AWS ?

   Il Runtime è accessibile tramite AWS Transform for mainframeToolbox on. AWS Transform for mainframe refactor

5. Quali sono i Runtime for AWS Transform supportati per le applicazioni mainframe refactored?

   AWS Transform for mainframeoffre un unico Runtime per soddisfare le diverse fasi del percorso di modernizzazione e le esigenze operative (vedi Runtime).AWSTransform for mainframe

6. Quando viene utilizzato AWS Transform for mainframe Runtime?

   Un AWS Transform for mainframe Runtime è necessario per supportare l'esecuzione di AWS Transform for mainframe Runtime. Durante i progetti di refactoring basati su AWS Transform for mainframe è necessario un runtime per testare le applicazioni refactorizzate. Una volta terminato il progetto di refactoring, è necessario anche un runtime per la manutenzione, il test e l'esecuzione di Transform for mainframe delle applicazioni refattorizzate in produzione. AWS

7. In che modo vengono AWS distribuite le nuove versioni di Transform for AWS mainframe Runtime?

   Le versioni AWS Transform for mainframe di Runtime sono accessibili tramite AWS Transform for mainframeToolbox. Consulta le note di rilascio di AWSTransform for mainframe.

8. Con che frequenza vengono rilasciate nuove versioni principali e secondarie di AWS Transform for mainframe runtime?

Per ulteriori informazioni, consulta Ciclo di vita dei componenti di modernizzazione del AWS mainframe.

9. In che modo AWS fornisce supporto per AWS Transform for mainframe Runtime?

   Il supporto viene fornito tramite Supporto AWS, laddove i problemi vengono risolti inviando un ticket, e si applica lo SLA standard. Per ulteriori informazioni, consulta Ciclo di vita dei componenti di modernizzazione AWS del mainframe.

10. Cosa comporta AWS Mainframe Modernization Transform for mainframe Runtime AWS ?

    AWS Transform for mainframe Runtime include librerie di strumenti per accelerare la modernizzazione, facilitare le integrazioni nel cloud e migliorare la qualità e la manutenibilità del codice. Inoltre, consente una maggiore automazione della modernizzazione facilitando le transizioni tra architetture legacy e architetture cloud. Il runtime fornisce supporto per la gestione di verbi e strutture di dati legacy, rappresentazioni della memoria utilizzando idiomi java. Consente di creare applicazioni modernizzate basate su tecniche di programmazione orientate agli oggetti e in grado di riprodurre flussi di controllo legacy. Modernizza i set di dati VSAM esistenti o il supporto dei database gerarchici IMS utilizzando un database relazionale come Amazon Aurora. Fornisce sostituti Java per le utilità di sistema legacy (IDCAMS, IEBGENER, DFSORT, ecc.) e i sistemi di gestione delle transazioni legacy (CICS, IMS). Facilita le integrazioni nel cloud con la memorizzazione nella cache in Amazon ElastiCache e il supporto per soluzioni di AWS messaggistica (SQS, Kinesis).

11. AWS Transform for mainframe Runtime supporta architetture di computer non x86?

    Attualmente, AWS Transform for mainframe Runtime supporta solo architetture e sistemi di calcolo basati su x86. AWS Transform for mainframe Runtime non supporta l'elaborazione basata su ARM e Graviton.

12. In che modo i clienti possono rimanere informati sulle versioni di AWS Transform for mainframe Runtime, comprese le notifiche di nuove versioni e l'accesso alla cronologia delle versioni e alle note di rilascio?

    Le nuove versioni di AWS Transform for mainframe Runtime vengono caricate sulla nostra pagina di rilascio ufficiale. Consigliamo di controllare regolarmente questa pagina, preferibilmente ogni 3 mesi, per le versioni e gli aggiornamenti più recenti. Per quanto riguarda l'accesso alla cronologia delle versioni e alle note di rilascio, la disponibilità dipende dalla data end-of-life (EOL) di ciascuna versione principale. Per informazioni dettagliate sulle date di EOL, sulla pianificazione

degli aggiornamenti delle versioni e sull'accesso alle informazioni cronologiche, consulta AWS Transform for mainframe lifecycle.

13. Quali sono i componenti principali dell'architettura di alto livello di AWS Transform for mainframe Runtime?

    L'architettura AWS Transform for mainframe Runtime comprende due tipi di componenti principali. Le prime sono le librerie Java (file jar) archiviate in una cartella condivisa (accessibile al classloader del server delle applicazioni) che forniscono supporto a costrutti e istruzioni legacy. In secondo luogo ci sono le applicazioni web (file war) contenenti applicazioni basate su Spring che forniscono framework e servizi a programmi modernizzati. Il runtime include anche: un registro dei programmi che raccoglie tutti i programmi per l'invocazione e le chiamate tra programmi e uno Scripts Registry che raccoglie tutti gli script di lavoro modernizzati. Questi componenti interagiscono per fornire un framework di ingresso ed esecuzione unificato basato su REST per applicazioni modernizzate. Il Runtime e l'applicazione modernizzata vengono distribuiti insieme in un server di applicazioni (ad esempio Tomcat).

14. Come configurare la cartella condivisa contenente gli artefatti di AWS Transform for mainframe Runtime?

    Gli artefatti (jar) di AWS Transform for mainframe Runtime devono essere raccolti in una cartella condivisa, accessibile al classloader del server delle applicazioni. Per un server tomcat, la configurazione viene effettuata modificando il normale file di configurazione denominato catalina.properties. Ad esempio, se hai creato la cartella condivisa come cartella denominata «shared», nella cartella tomcat, dovrai modificare la voce common.loader in catalina.properties per rendere la cartella condivisa accessibile al tomcat classloader, in quanto tale:

    ```
    common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.home}/
    lib","${catalina.home}/lib/*.jar","${catalina.home}/shared","${catalina.home}/shared/
    *.jar"
    ```

15. In che modo Transform for mainframe Runtime gestisce l'apolidia e la AWS gestione delle sessioni?

    AWS Transform for mainframe Runtime implementa l'assenza di stato e la gestione delle sessioni attraverso diversi meccanismi. Per le sessioni HTTP, utilizza l'identificazione basata su cookie con archiviazione cache esterna per il contesto dell'utente. Le sessioni possono essere archiviate in vari datastore tra cui Amazon ElastiCache, cluster Redis o mappe in memoria. Il design stateleless garantisce che la maggior parte degli stati non transitori venga archiviata esternamente in una «singola fonte di verità» comune, garantendo elevata disponibilità e scalabilità orizzontale.

Questo approccio, combinato con il bilanciamento del carico e le sessioni condivise, consente la distribuzione del dialogo rivolto all'utente su più nodi.

16. Che ruolo svolgono le applicazioni Web nell'ambiente AWS Transform for mainframe Runtime?

    Le applicazioni Web in AWS Transform for mainframe Runtime svolgono diverse funzioni chiave. Forniscono framework di esecuzione che riproducono ambienti e monitor delle transazioni legacy (come batch JCL, CICS, IMS). Offrono punti di ingresso basati su REST `gapwalk-application.war` per l'attivazione e il controllo di transazioni, programmi e batch. Inoltre, forniscono l'emulazione dei programmi forniti dal sistema operativo e dei programmi «driver» specializzati da cui dipendono le applicazioni legacy per accedere a servizi come IMS DB o alle finestre di dialogo utente tramite MFS.

17. Come vengono registrati e gestiti i programmi in Transform for mainframe Runtime? AWS

    I programmi in AWS Transform for mainframe Runtime vengono registrati tramite un ProgramRegistry sistema che si popola durante l'avvio del server. Ogni programma implementa l'interfaccia Program ed è contrassegnato come componente Spring. I programmi vengono registrati utilizzando i loro identificatori, con più voci possibili se un programma ha più identificatori. Il processo di registrazione è automatico e viene registrato nei log di Tomcat. ProgramRegistryConsente ad altri programmi e script di individuare e richiamare i programmi registrati, mantenendo la modularità e l'interconnettività del sistema modernizzato.

18. Come viene gestita la configurazione in AWS Transform for mainframe Runtime Applications?

    La configurazione in AWS Transform for mainframe Runtime è gestita tramite file YAML utilizzando le funzionalità del framework Spring Boot. Vengono utilizzati due file di configurazione principali: application-main.yml per la configurazione del framework e per le opzioni specifiche del client. `application-profile.yml` Il sistema segue la logica di precedenza di Spring, che consente l'override della configurazione in vari modi. È possibile fornire una configurazione aggiuntiva tramite JNDI per i database e i parametri della riga di comando, offrendo flessibilità nella gestione della configurazione. La configurazione dei logger viene eseguita utilizzando i file di configurazione logback xml.

19. Che ruolo svolgono i manager dei segreti nella configurazione di AWS Transform for mainframe Runtime?

    I gestori di segreti in AWS Transform for mainframe Runtime proteggono i dati di configurazione sensibili come le credenziali del database e le password della cache Redis. Consentono l'archiviazione di dati critici in forma AWS segreta e vi fanno riferimento nei file di configurazione YAML. Il sistema supporta diversi tipi di segreti, inclusi i segreti del database che popolano

automaticamente tutti i campi pertinenti e i segreti a password singola per le risorse protette da password. Questo approccio migliora la sicurezza mantenendo i dati sensibili separati dalla configurazione dell'applicazione.

20. In che modo gli sviluppatori possono scrivere i propri programmi compatibili con AWS Transform for mainframe Runtime?

    Gli sviluppatori possono creare programmi compatibili con AWS Transform for mainframe Runtime implementando l'interfaccia del programma e seguendo schemi specifici. Il programma deve essere dichiarato come componente Spring, implementare i metodi richiesti ed essere registrato correttamente in. ProgramRegistry Gli sviluppatori devono creare classi di contesto e configurazione complementari, gestire gli identificatori di programma e garantire una corretta integrazione con il framework Spring. L'implementazione deve seguire le convenzioni di AWS Transform for mainframe Runtime per la struttura e l'esecuzione del programma.

21. In che modo AWS Transform for mainframe Runtime gestisce gli errori di esecuzione del programma?

    AWS Transform for mainframe Runtime gestisce gli errori di esecuzione del programma attraverso diversi meccanismi. Per i lavori in batch, acquisisce lo stato di esecuzione, i codici di uscita e informazioni dettagliate sugli errori nei dettagli di esecuzione del lavoro. La gestione degli errori include codici di uscita specifici (-1 per errori tecnici, -2 per errori dei programmi di servizio) e registrazioni dettagliate nei log di Tomcat. Il sistema può essere configurato per ripristinare le transazioni in base alle eccezioni di runtime e offre opzioni per la notifica e il ripristino degli errori. I dettagli degli errori sono accessibili tramite gli endpoint REST per il monitoraggio e la risoluzione dei problemi.

22. Quali funzionalità di monitoraggio di AWS Transform for mainframe Runtime sono disponibili per i lavori in batch?

    AWS Transform for mainframe Runtime offre funzionalità di monitoraggio per i processi in batch attraverso vari endpoint. Tiene traccia dello stato di esecuzione del lavoro, dei start/end tempi, della modalità di esecuzione e dei risultati dettagliati. Il sistema offre endpoint per elencare gli script attivati, recuperare i dettagli sull'esecuzione dei lavori e monitorare i lavori attualmente in esecuzione. Gli endpoint di Metrics forniscono statistiche JVM, conteggi delle sessioni e metriche dettagliate sull'esecuzione dei batch. La piattaforma supporta anche l'impaginazione e il filtraggio temporale dei dati di monitoraggio.

23. In che modo vengono tracciati e gestiti gli stati di esecuzione dei job di AWS Transform for mainframe Runtime?

Gli stati di esecuzione del lavoro vengono tracciati tramite un sistema di stati completo che include stati come DONE, TRIGGERED, RUNNING, KILLED e FAILED. Ogni esecuzione di lavoro riceve un identificatore univoco per il tracciamento e conserva informazioni dettagliate sull'esecuzione, tra cui ora di inizio, ora di fine, informazioni sul chiamante e risultati di esecuzione. Il sistema fornisce endpoint REST per interrogare lo stato dei processi, gestire i lavori in esecuzione e recuperare la cronologia di esecuzione. Le informazioni sullo stato persistono nella memoria del server e possono essere eliminate in base all'età per la gestione delle risorse.

24. In che modo AWS Transform for mainframe Runtime gestisce le interazioni con i sistemi esterni?

    Il runtime gestisce le interazioni esterne del sistema attraverso vari meccanismi, inclusi gli endpoint REST per l'integrazione dei servizi, il supporto per le code di messaggi (SQS, RabbitMQ, IBM MQ) e le opzioni di connettività del database. Fornisce l'emulazione delle interazioni dei sistemi legacy tramite componenti specializzati, supporta comunicazioni sicure e include funzionalità SSL/TLS per la gestione di file system esterni. Il sistema supporta anche l'integrazione con provider di autenticazione esterni e può essere configurato per interagire con vari servizi di terze parti.

25. Come viene gestita l'autenticazione in AWS Transform for mainframe Runtime?

    AWS Transform for mainframe Runtime supporta più metodi di autenticazione ed è il OAuth2 meccanismo principale. Può integrarsi con provider di identità come Amazon Cognito o Keycloak. La configurazione dell'autenticazione è gestita tramite il file di configurazione principale denominato application-main.yml, in cui è possibile definire impostazioni di sicurezza, provider di identità e metodi di autenticazione. Il sistema supporta funzionalità come la protezione XSS, CORS, CSRF e può essere configurato sia per la sicurezza globale che per la sicurezza degli endpoint specifici. Per lo sviluppo, è disponibile anche un sistema di autenticazione locale con credenziali super admin predefinite.

26. In che modo AWS Transform for mainframe Runtime garantisce un'elevata disponibilità?

    AWS Transform for mainframe Runtime garantisce un'elevata disponibilità attraverso diversi meccanismi. Implementa l'assenza di stato archiviando gli stati non transitori in uno storage condiviso esterno, consentendo a più istanze di applicazioni di lavorare insieme. Il sistema supporta il bilanciamento del carico e le sessioni condivise, consentendo la distribuzione delle richieste su più nodi. Per l'archiviazione dei dati, può utilizzare database e sistemi di caching ad alta disponibilità. L'architettura supporta il failover automatico e può essere implementata su più zone di disponibilità per una maggiore affidabilità.

27. Quale componente viene utilizzato per riprodurre le transazioni distribuite CICS con AWS Transform for mainframe?

AWS Transform for mainframe Runtime fornisce un endpoint dedicato per consentire di richiamare le transazioni JICS esistenti come parte di una transazione globale (supporto XA). Il supporto del commit in due fasi sottostanti si basa sul componente software Atomikos.

28. Qual è il nome AWS Transform for mainframe delle classi utilizzate per definire il comportamento specifico del programma?

    Ogni programma è associato a una classe di configurazione dedicata che consente di specificare i comportamenti specifici del programma. Per ulteriori informazioni sulle convenzioni di denominazione e localizzazione, vedete AWS Transform for mainframe structure of modernized application

29. Quale codifica ha il seguente ordine di sequenza di caratteri: spazio, caratteri minuscoli, caratteri maiuscoli, numeri?

    Set di caratteri appartenenti alla famiglia di varianti EBCDIC (come, ecc.). CP1047 CP297

30. Quali sono le dimensioni dei prezzi di AWS Transform for mainframe Runtime?

    AWS Mainframe Modernization-core-hours (vedi i prezzi di AWS Mainframe Modernization).

31. Qual è il meccanismo utilizzato per trasferire i dati grezzi tramite HTTP agli endpoint del programma?

    Stringhe codificate in Base64.

32. In che modo un utente avvia un processo in batch?

    Utilizzo di una chiamata HTTP a uno degli endpoint batch dedicati (vedere la pagina della documentazione sugli endpoint batch).

33. Quale endpoint AWS Transform for mainframe Runtime è il punto di ingresso principale dell'applicazione front-end web principale?

    ```
    /transaction
    ```

34. Cosa significa AWS Transform for mainframe JICS?

    AWS Transform for mainframe JICS è il componente di runtime utilizzato per supportare la modernizzazione delle risorse CICS. Le definizioni delle risorse sono archiviate in un archivio dati dedicato. Per amministrarle, utilizza l'API REST o la console dell'applicazione JICS. Per informazioni, consulta Gestire la console dell'applicazione JICS in AWS Transform for mainframe.

35. Quali meccanismi di caching di AWS Transform for mainframe Runtime sono disponibili?

AWS Transform for mainframe Runtime supporta diversi meccanismi di memorizzazione nella cache, tra cui Redis e. EhCache Redis è consigliato per gli ambienti di produzione, poiché fornisce una cache persistente condivisa su più nodi. EhCache è disponibile per implementazioni autonome con caching locale volatile incorporato. Il sistema supporta la memorizzazione nella cache di vari componenti, tra cui Blusam dati, informazioni sulla sessione, risorse JICS e code di archiviazione temporanee. La configurazione della cache può essere personalizzata per diversi casi d'uso e requisiti di prestazioni.

36. Come possiamo stimare il prezzo di una implementazione AWS Mainframe Modernization AWS Transform for mainframe Runtime?

    AWS fornisce stime ai clienti in base ai loro requisiti e all'architettura di destinazione.

37. Qual è il prezzo di AWS Mainframe Modernization AWS Transform for mainframe Runtime?

    AWS Mainframe Modernization offre due modelli di prezzo per AWS Transform for mainframe: un'opzione Managed Runtime che include il runtime, le risorse di calcolo, lo storage interno e l'automazione e un'opzione di runtime non gestita che copre solo il runtime AWS stesso. Transform for mainframe Per le AWS implementazioni, entrambe utilizzano una struttura tariffaria. pay-as-you-go Per informazioni più up-to-date dettagliate sui prezzi, si consiglia di consultare la pagina ufficiale dei prezzi di AWS Mainframe Modernization.

38. E se fosse necessario distribuire un'applicazione refactorizzata AWS Transform for mainframe su un'infrastruttura non elencata nel runtime supportato?

    Se è necessario distribuire un'applicazione refattorizzata AWS Transform for mainframe su un'infrastruttura non elencata nel runtime supportato, sono disponibili diverse opzioni. Innanzitutto, verifica se la tua infrastruttura è compatibile con le opzioni di distribuzione esistenti come Amazon EKS Anywhere o altre piattaforme di orchestrazione dei container. In tal caso, potresti essere in grado di utilizzare AWS Transform for mainframe Runtime. Per le infrastrutture non compatibili, consigliamo di consultare uno specialista di AWS mainframe per esplorare soluzioni personalizzate o potenziali adattamenti. Puoi anche inviare una Product Feature Request (PFR) per un supporto esteso dell'infrastruttura. Contattate il vostro AWS rappresentante per discutere delle vostre esigenze specifiche e dell'approccio migliore per il vostro ambiente.

39. Come viene concesso in licenza AWS Transform for mainframe Runtime? È open source?

    AWS Transform for mainframe Runtime non è open source. È distribuito tramite AWS IP come servizio nativo del cloud. Esistono due opzioni di implementazione:

a. AWS Transform for mainframe Managed, il runtime viene distribuito in un servizio AWS gestito dedicato, sfruttando un ambiente completamente preconfigurato e pronto per l'implementazione senza configurazione né amministrazione.

b. AWS Transform for mainframe Non Managed, che può essere implementato nella tua AWS architettura personalizzata basata su Amazon EC2 o Amazon ECS/AWS Fargate, che devi fornire e configurare autonomamente. Entrambe le opzioni prevedono costi di runtime, che sono inclusi nelle stime del progetto che ti vengono fornite. Poiché si tratta di un servizio gestito con Supporto accesso, non è necessario il codice sorgente. Per maggiori dettagli sui prezzi, consulta la pagina dei prezzi per la modernizzazione del AWS mainframe.

40. Come vengono gestite le modifiche e gli aggiornamenti ai framework e alle AWS librerie Transform for mainframe?

    AWS I framework e le librerie Transform for mainframe vengono aggiornati tramite regolari processi di generazione e distribuzione del codice. Questi aggiornamenti sono gestiti come parte del ciclo di vita della modernizzazione del AWS mainframe, che include aggiornamenti di versione e supporto da parte del team Transform for mainframe AWS o dei partner certificati. Per informazioni dettagliate sul controllo delle versioni, sui processi di aggiornamento e sulle tempistiche di supporto, consulta la documentazione sul ciclo di vita della modernizzazione del mainframe.AWS

41. Quali sono le versioni supportate degli strumenti (Tomcat, Postgres, MQ, ecc.) e delle dipendenze (Spring, Angular, ecc.) utilizzate da AWS Transform for mainframe Runtime?

    Vedi i dettagli nelle note di rilascio.

42. Cosa significa «Standalone» nel contesto di BAC e JAC?

    Standalone si riferisce a una speciale modalità di imballaggio e distribuzione per BAC (BlusamAdministration Console) e JAC (JICS Administration Console) che consente a queste applicazioni Web di funzionare indipendentemente nel proprio server Tomcat, separato dall'applicazione modernizzata. Le versioni standalone BAC e JAC sono disponibili in. `aws-bluage-webapps-x.y.z.zip` Le versioni non standalone BAC e JAC sono disponibili nella cartella Under. `gapwalk-x.y.z.zip` `webapps-consoles` Vedi gli artefatti di AWS Transform for mainframe Runtime.

Dati

1. Quali opzioni di database sono disponibili per le applicazioni modernizzate, per quanto riguarda la modernizzazione del database legacy?

   Le applicazioni modernizzate possono utilizzare diverse opzioni di database moderne, tra cui: PostgreSQL, Amazon Aurora, RDS per PostgreSQL, database Oracle, MS-SQL e IBM Db2. Queste opzioni offrono flessibilità nella scelta del sistema di database più appropriato in base a requisiti specifici, sfruttando al contempo i vantaggi dei moderni sistemi di gestione dei database e delle funzionalità native del cloud.
2. Qual è la copertura di trasformazione di IBM Db2 in Postgres DDL? z/OS

   Trasformazione completa (compresi i vincoli del database).
3. AWS Transform for mainframe supporta la generazione di dati di gruppo (GDG)?

   Sì, è supportato l'utilizzo di GDG in batch, con il supporto di generazioni relative e assolute e di strategie di pulizia automatica.
4. AWS Transform for mainframe supporta set di dati concatenati?

   Sì, è supportato l'utilizzo di set di dati concatenati in batch. Con la concatenazione in azione, diversi set di dati possono essere letti come un unico set di dati. Tieni presente che i set di Blusam dati non possono far parte di una concatenazione.
5. Qual è il processo applicato alle query SQL?

   Modificato durante la trasformazione del codice, a seconda del database di destinazione.
6. Quali opzioni si applicano se ci sono più database per un'applicazione?

   Configura il database di destinazione per ogni query e definisci tutti i database nell'applicazione e in Apache Tomcat.
7. Può Blusam essere disabilitato?

   Sì, nel file di configurazione principale e non è richiesto alcun database (per ulteriori informazioni, consulta la pagina della documentazione di Blusam configurazione).
8. Quale API AWS Transform for mainframe viene utilizzata per sostituire database come IMS DB?

   L'API JHDB (Java DataBase Hierarchical).
9. Quale prodotto AWS Transform for mainframe può essere utilizzato per migrare dati e database legacy verso un moderno sistema di gestione di database relazionali (RDBMS)?

AWS Strumento di modernizzazione del database Transform for mainframe (Data Migrator).

10. Cos'è AWS Transform for mainframe Data Simplifier e quale problema risolve durante la modernizzazione?

    Data Simplifier è una libreria di base di AWS Transform per mainframe che affronta la sfida della gestione dei modelli di accesso alla memoria legacy in Java. Fornisce costrutti per supportare l'accesso alla memoria di basso livello, i tipi di dati legacy (ad esempio suddivisi in zone, compressi, alfanumerici) e l'accesso alla structured/raw memoria mista, comuni nelle applicazioni mainframe ma non disponibili in modo nativo in Java. La libreria espone queste funzionalità attraverso modelli Java familiari, come quelli basati su classi APIs, rendendole getters/setters accessibili agli sviluppatori Java pur mantenendo le funzionalità legacy.

11. In che modo AWS Transform for mainframe gestisce i layout di memoria e le strutture di dati legacy?

    AWS Transform for mainframe gestisce i layout di memoria legacy tramite l'interfaccia Record, che fornisce un'astrazione di array di byte con dimensioni fisse. Per dati strutturati come COBOL «01 data item», utilizza sottoclassi generate automaticamente durante la modernizzazione. RecordEntity Queste classi mantengono la struttura gerarchica dei dati legacy, con ogni elemento che ha una relazione padre-figlio. Il sistema supporta sia l'accesso alla memoria grezza che i modelli di accesso strutturati, preservando la flessibilità dei sistemi legacy e fornendo al contempo un'interfaccia di programmazione moderna.

12. In che modo AWS Transform for mainframe gestisce la modernizzazione dei set di dati VSAM?

    Il componente Blusam fornisce il supporto per la modernizzazione dei set di dati VSAM, con un'API dedicata, endpoint e un'applicazione web di amministrazione (BAC: Blusam Administration Console). Blusam si affida a un database relazionale come backend (PostgreSQL, utilizzando RDS o Aurora).

Trasformazione

1. Dove posso trovare dettagli sul processo di trasformazione?

   Consulta la documentazione di AWS Transform for mainframe refactor.

2. Quali sono i nomi dei moduli generati da AWS Transform for mainframe?

   Servizio, entità, web e strumenti.

3. Perché è stata Java/Spring scelta come una delle tecnologie di destinazione per AWS Transform for mainframe?

   Java/Spring è stata scelta come tecnologia di destinazione per la sua adozione diffusa, l'ampio pool di talenti e le solide capacità aziendali. L'ecosistema Java offre ampie librerie, framework e strumenti che supportano lo sviluppo di applicazioni moderne. Il framework Spring offre funzionalità di livello aziendale, funzionalità native del cloud e segue le migliori pratiche del settore, rendendolo ideale per le applicazioni modernizzate.

4. Qual è il nome del progetto principale che contiene i moduli generati da Transform for mainframe? AWS

   Il nome del progetto principale ha il suffisso «-pom» e può essere definito nel Transformation Center utilizzando la proprietà Transform denominata project.

5. In che modo AWS Transform for mainframe gestisce la modernizzazione dello scheduler legacy, se fornita?

   Le risorse di pianificazione legacy non vengono modernizzate da Transform for mainframe. AWS Vengono presi in considerazione durante la fase di valutazione, per aiutare a identificare possibili artefatti mancanti.

6. Qual è il requisito per eseguire il debug del codice generato con Transform for mainframe? AWS

   Qualsiasi ambiente di sviluppo integrato (IDE) che supporti Java, come Eclipse o. JetBrain VisualCode

Implementazione

1. Quali ambienti sono disponibili per implementare l'applicazione modernizzata con AWS Transform for mainframe?

   Windows Server, server Linux e contenitore Docker Linux.

2. Le applicazioni refactorizzate AWS Transform for mainframe possono essere eseguite su qualsiasi infrastruttura?

   Sebbene le applicazioni refattorizzate AWS Transform for mainframe non siano progettate per funzionare su alcuna infrastruttura, offrono una notevole flessibilità nelle opzioni di implementazione. Queste applicazioni possono essere implementate su varie piattaforme di elaborazione, tra cui servizi gestiti sul cloud, elaborazione senza server e infrastruttura locale.

3. Quale configurazione MQ supporta AWS Transform for mainframe?

   SQS, IBM MQ WebSphere .

4. In quali server applicativi un utente può implementare la logica delle applicazioni aziendali Java con il runtime AWS Mainframe Modernization?

   Apache Tomcat, versione successiva o uguale alla 10.1.

5. In che modo l'applicazione refactorizzata si integra con altre applicazioni Servizi AWS come Amazon Aurora?

   L'applicazione modernizzata si integra Servizi AWS supportando la trasformazione verso soluzioni di database native del cloud come Amazon Aurora e RDS per PostgreSQL. AWS Transform for mainframe garantisce l'integrazione tra applicazioni modernizzate e consente alle organizzazioni di utilizzare le funzionalità del cloud. Servizi AWS Questa integrazione si estende sia all'archiviazione dei dati che ai servizi applicativi all'interno dell' AWS ecosistema. Oltre allo storage del database, AWS Transform for mainframe Runtime si integra con vari sistemi, tra cui Servizi AWS Amazon ElastiCache for Redis caching, per la gestione della configurazione e AWS Mainframe Modernization Gestione dei segreti AWS per l'implementazione. Supporta Amazon EC2, Amazon EKS, ECS gestiti da Fargate per l'implementazione di container. Il sistema può utilizzare AWS Identity and Access Management per l'autenticazione Amazon Simple Storage Service per lo storage e supporta l'integrazione con altri Servizi AWS tramite connettori di configurazione e servizio.

6. In che modo l'applicazione rifattorizzata garantisce il rispetto dei requisiti di scalabilità?

   La soluzione garantisce la scalabilità trasformando le applicazioni in architetture native del cloud che possono utilizzare l'infrastruttura elastica. AWS Implementa modelli di progettazione moderni e best practice che consentono la scalabilità orizzontale e verticale. L'approccio orientato ai servizi consente la scalabilità indipendente dei componenti. Le applicazioni modernizzate possono sfruttare le caratteristiche di scalabilità intrinseche dei servizi cloud.

7. Cosa succede dopo il completamento del refactoring del codice sorgente?

   Dopo il refactoring del codice sorgente, si verificano due passaggi principali. Innanzitutto, viene creata l'applicazione refactorizzata. In secondo luogo, l'applicazione viene distribuita. L'implementazione viene eseguita nel tuo Account AWS (AWS Mainframe Modernization AWSTransform for mainframe Runtime), dove i clienti gestiscono la propria infrastruttura e la implementano su varie piattaforme, tra cui Amazon EC2, ECS su EC2 o Fargate, EKS su EC2.

8. Come posso implementare ed eseguire un'applicazione modernizzata con AWS Transform for mainframe su un'AMI Amazon Linux personalizzata?

   Ciò può essere ottenuto distribuendo l'applicazione utilizzando AWS Transform for mainframe Runtime su Amazon EC2. Il processo prevede la creazione di un'applicazione Java/Spring con una dipendenza dalla libreria AWS Transform for mainframe Runtime e la sua distribuzione su un'AMI Amazon Linux personalizzata. Per istruzioni dettagliate su questo approccio, consulta Configurare AWS Transform for mainframe Runtime su Amazon EC2.

9. È disponibile un'Amazon Machine Image (AMI)? È disponibile un'immagine Docker?

   - AMI: No, a causa delle esigenze dei clienti di personalizzare e configurare il proprio ambiente come preferiscono, non è disponibile alcuna AMI. I clienti possono recuperare gli artefatti AWS Transform for mainframe e configurare la propria istanza in base alle proprie esigenze.
   - Docker Image: No, non è disponibile un'immagine docker pronta all'uso, ma la pagina AWS Configura Transform for mainframe Runtime on container spiega come creare e distribuire la propria immagine docker AWS basata sui binari di Transform for mainframe Runtime, su un sistema di gestione dei container adatto.

10. Il cliente può impacchettare ed eseguire un'applicazione AWS Transform for mainframe come contenitore Docker?

    Sì, fai riferimento a Configurare AWS Transform for mainframe Runtime sul contenitore.

11. Come funziona la pianificazione dei lavori con batch?

    È integrato con il ramo Control-M/Stone o con qualsiasi altro programma di pianificazione distribuito.

Sicurezza

1. In che modo l'applicazione protegge dagli attacchi di SQL injection?

   L'applicazione implementa dappertutto le best practice SQL. Tutte le query del database utilizzano istruzioni preparate, che prevengono efficacemente gli attacchi di SQL injection separando il codice SQL dai dati forniti dall'utente. Ciò garantisce che l'input dell'utente venga sempre trattato come dati anziché come codice eseguibile.

2. Quali misure sono in atto per prevenire le vulnerabilità di iniezione dei comandi del sistema operativo?

AWS i team addetti alla sicurezza effettuano revisioni periodiche della sicurezza di AWS Transform for mainframe Runtime. Se i team rilevano anomalie, ad esempio del codice che potrebbe consentire l'iniezione di comandi nel sistema operativo, si adoperano AWS immediatamente per risolverle. Questo monitoraggio continuo garantisce che il runtime rimanga sicuro contro le minacce di iniezione di comandi.

3. L'applicazione è vulnerabile agli attacchi che attraversano le directory?

   No. Questa vulnerabilità non si applica all'applicazione Angular front-end. Per il back-end, l'applicazione utilizza solo un endpoint con un contratto di interfaccia limitato che non contiene mai informazioni sul «percorso» o sulla «directory». Questo design elimina il rischio di attacchi che attraversano le directory.

4. In che modo l'applicazione protegge dagli attacchi XSS?

   L'applicazione segue le migliori pratiche di sicurezza di Angular come descritto nella documentazione ufficiale sulla sicurezza di Angular. La natura dell'applicazione è intrinsecamente limitata in termini di interfaccia utente poiché corrisponde all'applicazione mainframe legacy. L'applicazione non URLs gestisce script dinamici o codice HTML generato dal back-end. Questa superficie di attacco limitata riduce significativamente i rischi XSS.

5. Quale protezione viene implementata contro gli attacchi CSRF?

   L'applicazione utilizza il supporto CSRF nativo di Spring, che fornisce una protezione efficace contro gli attacchi di falsificazione delle richieste tra siti pronti all'uso.

6. L'input dell'utente può portare all'iniezione di intestazioni HTTP?

   No. La lunghezza degli input degli utenti è strettamente limitata in modo da corrispondere al formato delle applicazioni mainframe precedenti. Quando il back-end riceve una richiesta, formatta immediatamente l'input in base a formati predefiniti, in modo da troncare qualsiasi contenuto iniettato. Inoltre, l'applicazione non crea mai intestazioni HTTP in base agli input dell'utente nel singolo endpoint utilizzato dall'applicazione, rendendo impossibile l'iniezione di intestazioni HTTP.

7. In che modo l'applicazione è protetta dagli attacchi di clickjacking?

   L'applicazione esegue la maggior parte delle azioni utilizzando sequenze di tasti anziché clic del mouse, il che la rende intrinsecamente resistente agli attacchi di clickjacking. Questa scelta progettuale riduce in modo significativo la superficie di attacco per questo tipo di vulnerabilità.

8. Possono verificarsi attacchi di buffer overflow nell'applicazione?

No. La lunghezza degli input degli utenti è strettamente limitata per corrispondere al formato applicativo mainframe legacy. Quando il back-end riceve una richiesta, formatta immediatamente l'input in base ai formati corrispondenti, evitando che si verifichino condizioni di buffer overflow.

9. Come vengono gestiti il controllo e l'autorizzazione degli accessi?

   Il controllo degli accessi determina a livello globale se gli utenti hanno i diritti di accesso all'endpoint back-end unico. Una volta autenticati, il codice dell'applicazione e i dati gestiscono i controlli di autorizzazione. Ciò garantisce un'adeguata separazione delle preoccupazioni e una gestione sicura degli accessi.

10. Quali misure impediscono il dirottamento della sessione?

    L'uso combinato delle funzionalità native di Spring e Angular per la gestione delle sessioni e la prevenzione XSS, insieme all'adesione alle migliori pratiche del framework, impedisce il dirottamento delle sessioni. L'applicazione può migliorare ulteriormente la protezione mediante:

    - Utilizzo di HTTPS: assicura che tutte le comunicazioni siano crittografate.
    - Implementazione dell'autenticazione basata su token: l'applicazione può attivarla quando necessario.
    - Monitoraggio delle sessioni utente: l'applicazione può attivare il monitoraggio delle sessioni e il rilevamento delle anomalie per identificare attività sospette.

    Queste misure di sicurezza a più livelli forniscono una protezione completa contro i tentativi di dirottamento delle sessioni.

Ripiattaforma delle applicazioni con Rocket Software (precedentemente Micro Focus)

Questa guida descrive il end-to-end processo di ripiattaforma delle applicazioni mainframe utilizzando le soluzioni di modernizzazione del AWS mainframe su. AWS Descrive tutte le attività e include informazioni sulla configurazione e il funzionamento del runtime di modernizzazione del AWS mainframe su Amazon EC2, dalla configurazione e analisi iniziali alla creazione, al test e alla distribuzione delle applicazioni modernizzate su. AWS Tratta anche argomenti avanzati come l'utilizzo di strutture di dati legacy, l'utilizzo di modelli e progetti predefiniti e la configurazione dell'automazione per le sessioni di streaming.

Argomenti

- Configurare Rocket Software (precedentemente Micro Focus) (su Amazon EC2)
- Configurazione dell'automazione per le sessioni di streaming di Rocket Enterprise Analyzer (precedentemente Micro Focus) e Rocket Enterprise Developer
- Visualizza i set di dati come tabelle e colonne in Rocket Enterprise Developer (precedentemente Micro Focus Enterprise Developer)
- Modifica i set di dati utilizzando gli strumenti per i file di dati di Rocket Software (precedentemente Micro Focus) in Enterprise Developer
- Tutorial per Rocket Software (in precedenza Micro Focus)
- Utilità batch disponibili in Mainframe AWS Modernization

Configurare Rocket Software (precedentemente Micro Focus) (su Amazon EC2)

Modernizzazione del mainframe AWS fornisce diverse Amazon Machine Images (AMI) che includono prodotti con licenza Rocket Software (precedentemente Micro Focus). Queste AMI consentono di effettuare rapidamente il provisioning di istanze Amazon Elastic Compute Cloud (Amazon EC2) per supportare gli ambienti Rocket Software che controlli e gestisci. Questo argomento fornisce i passaggi necessari per accedervi e avviarli. AMIs Il loro utilizzo AMIs è del tutto facoltativo e non è necessario per completare i tutorial di questa guida per l'utente.

Argomenti

- Prerequisiti per la configurazione di Rocket Software (precedentemente Micro Focus) (su Amazon EC2)
- Crea l'endpoint Amazon VPC per Amazon S3
- Richiedi l'aggiornamento della lista consentita per l'account
- Crea il AWS Identity and Access Management ruolo
- Concedi a License Manager le autorizzazioni richieste
- Iscriviti ad Amazon Machine Images
- Avvia un'istanza di Modernizzazione del mainframe AWS Rocket Software (precedentemente Micro Focus)
- Subnet o VPC senza accesso a Internet

Prerequisiti per la configurazione di Rocket Software (precedentemente Micro Focus) (su Amazon EC2)

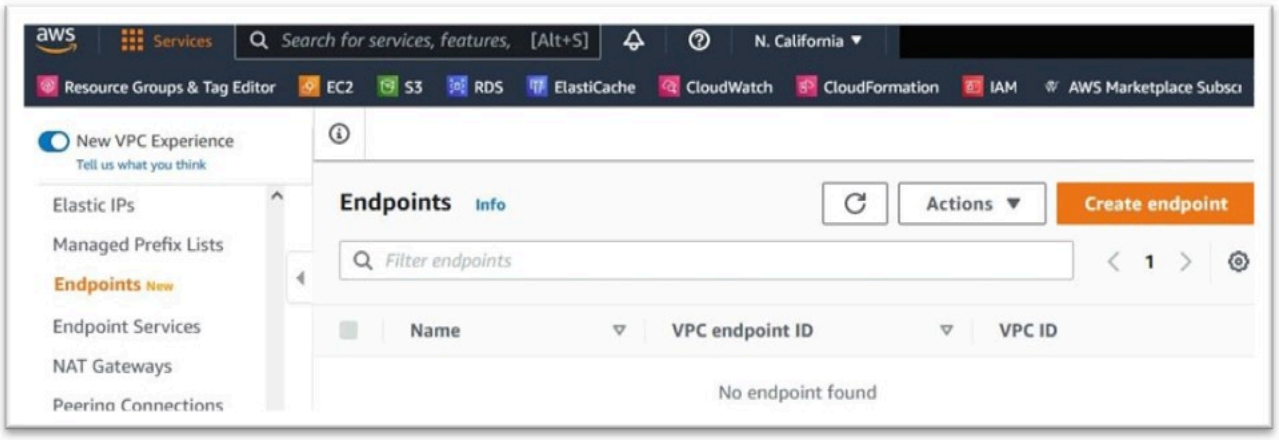
Quando configuri Rocket Software (su Amazon EC2), assicurati di soddisfare i seguenti prerequisiti.

- 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 Modernizzazione del mainframe AWS servizio. Vedi AWS Servizi per regione. 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.

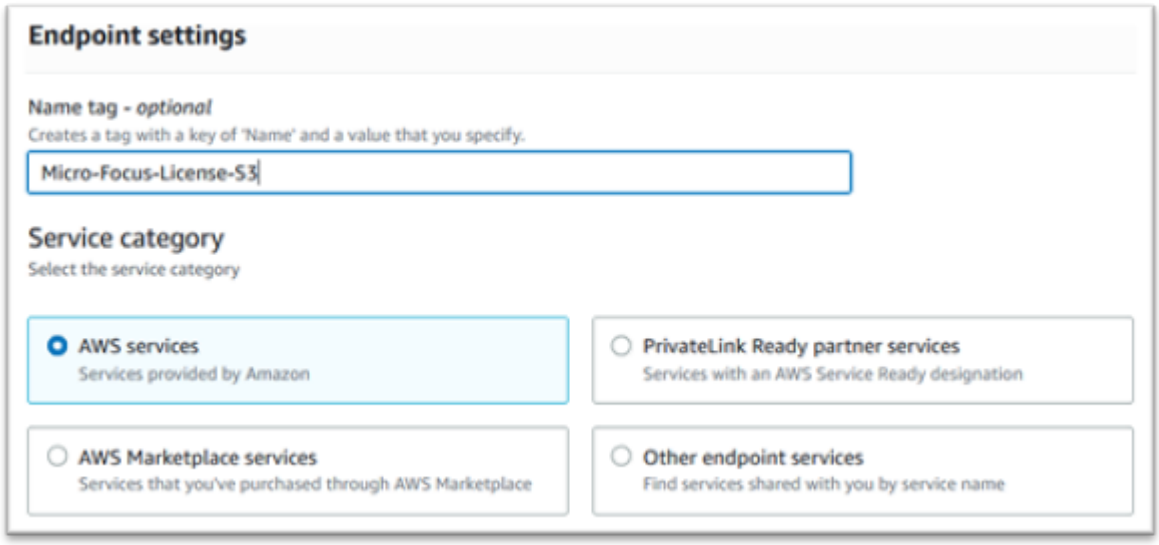
Crea l'endpoint Amazon VPC per Amazon S3

In questa sezione, crei un endpoint Amazon VPC da utilizzare con Amazon S3. La configurazione di questo endpoint ti aiuterà in seguito a configurare l'accesso a Internet per VPC.

1. Accedi ad Amazon VPC in. Console di gestione AWS
2. Nel pannello di navigazione, seleziona Endpoints (Endpoint).
3. Seleziona Crea endpoint.

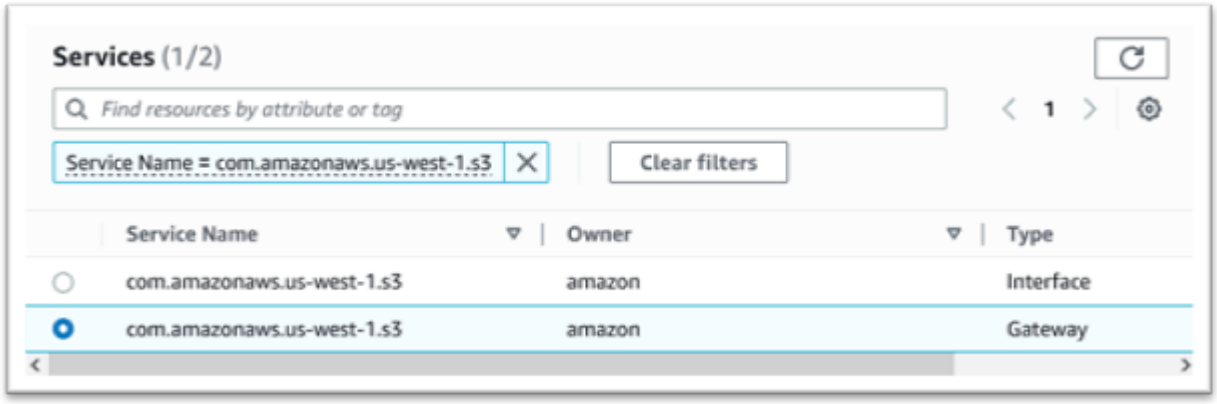


4. Inserisci un tag con il nome significativo, ad esempio: «Micro-Focus-License-S3".
5. Scegli i servizi AWS come categoria di servizi.

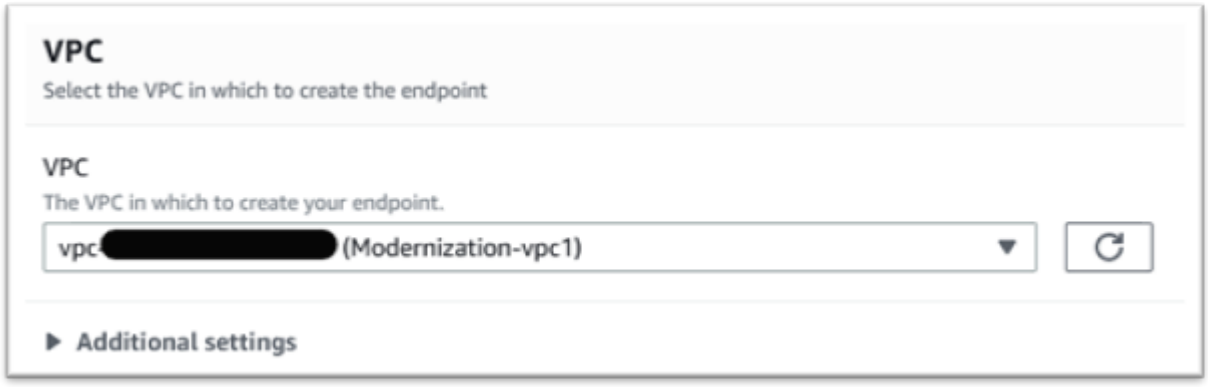


6. In Servizi cerca il servizio Amazon S3 Gateway: com.amazonaws. [regione] .s3.

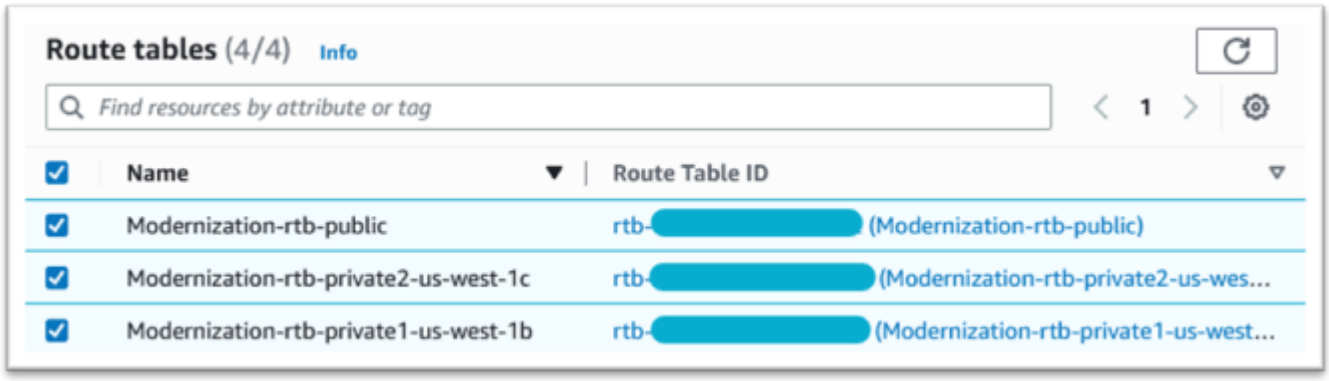
   Per `us-west-1` questo sarebbe:. `com.amazonaws.us-west-1.s3`
7. Scegli il servizio Gateway.



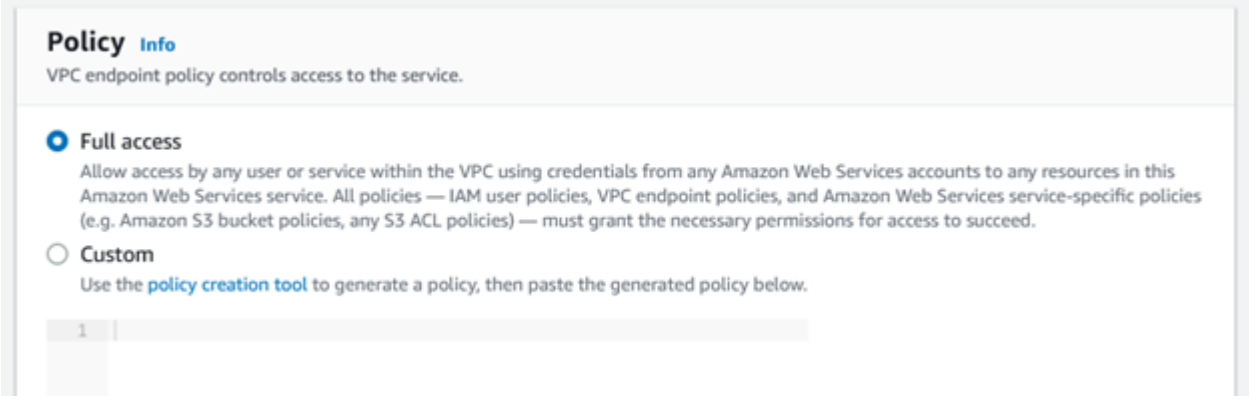
8. Per VPC scegli il VPC che utilizzerai.



9. Scegli tutte le tabelle di routing per il VPC.



10. In Politica scegli Accesso completo.



> **Note**
>
> Se decidi di creare una policy personalizzata, assicurati che abbia accesso al bucket Amazon S3. `s3://aws-supernova-marketplace-<region>-prod`

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 autorizzata per. Modernizzazione del mainframe AWS AMIs 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 per tutte le istanze Rocket Software 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**
>
> Contatta il tuo AWS rappresentante o Supporto AWS chi aprirà il ticket di assistenza per la richiesta Allowlist per tuo conto. Non può essere richiesto direttamente da te e il completamento della richiesta potrebbe richiedere diversi giorni.

Crea il AWS Identity and Access Management ruolo

Crea una AWS Identity and Access Management policy e un ruolo da utilizzare per le istanze Modernizzazione del mainframe AWS Amazon EC2. 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 software Rocket. Per ulteriori informazioni sui profili di istanza, consulta Usare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su istanze Amazon EC2.

Creazione di una policy IAM

Una policy IAM viene prima creata e poi associata al ruolo.

1. Passa AWS Identity and Access Management a Console di gestione AWS.
2. Scegli Politiche e poi Crea politica.



3. Scegli la scheda JSON.



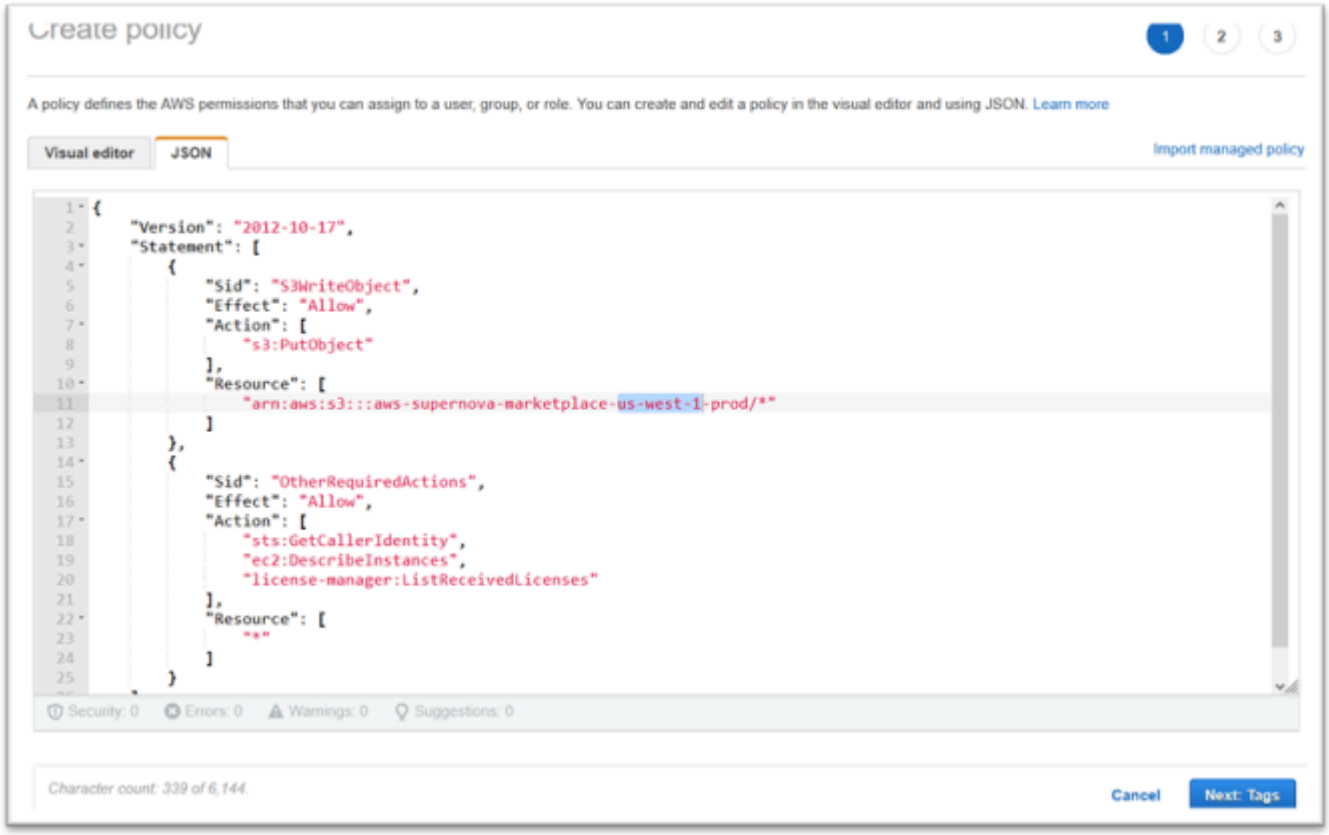
4. Sostituisci `us-west-1` il codice JSON seguente con quello Regione AWS in cui è stato definito l'endpoint Amazon S3, quindi copia e incolla il codice JSON nell'editor delle policy.

   JSON

```
{
    "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": [
                "*"
```

```
            ]
        }
    ]
}
```

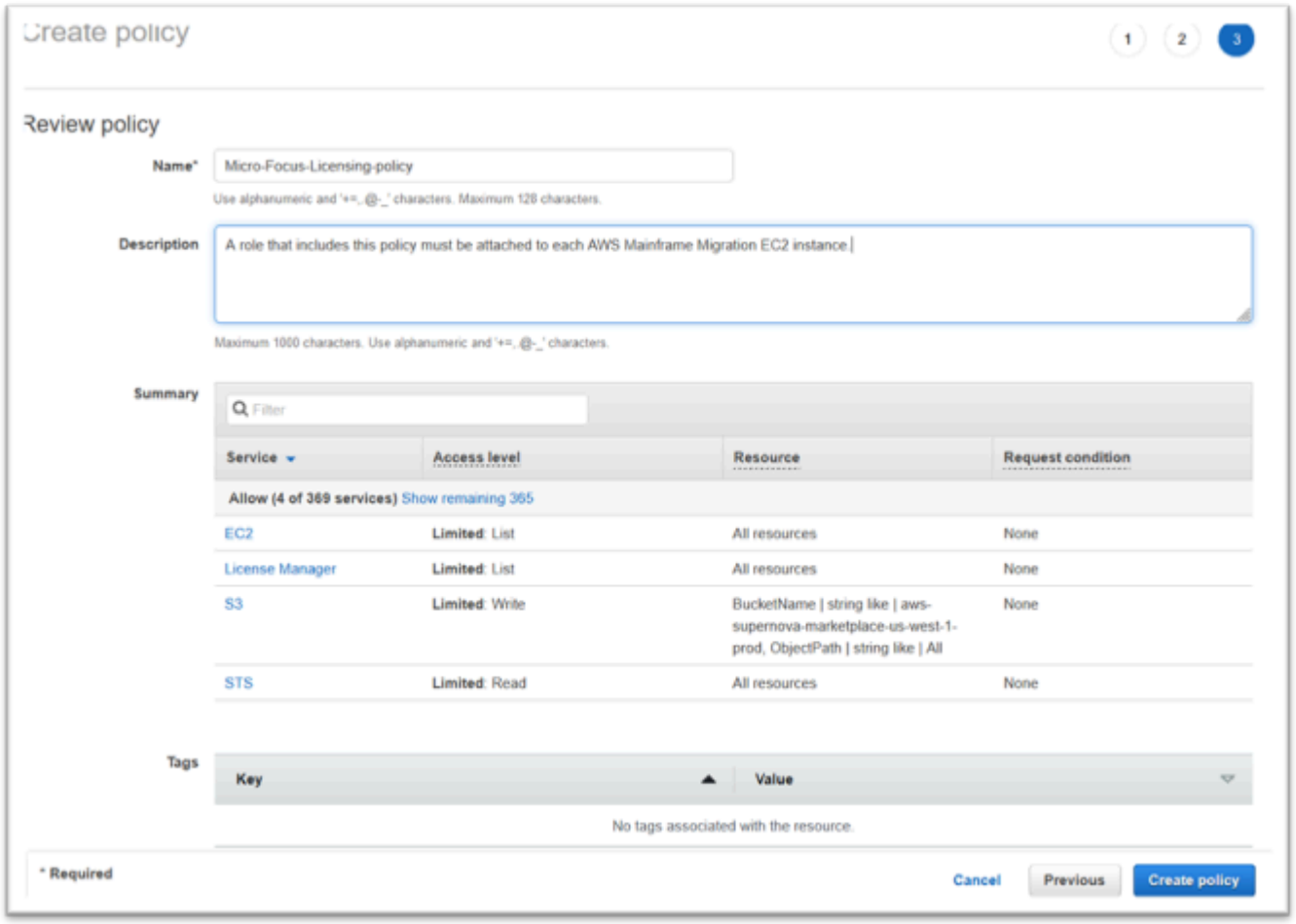
> **Note**
>
> Le azioni sotto il Sid `OtherRequiredActions` non supportano le autorizzazioni a livello di risorsa e devono essere specificate nell'elemento resource. *



5. Scegli Successivo: Tag.



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 politica deve essere collegato a ciascuna istanza Modernizzazione del mainframe AWS Amazon EC2".

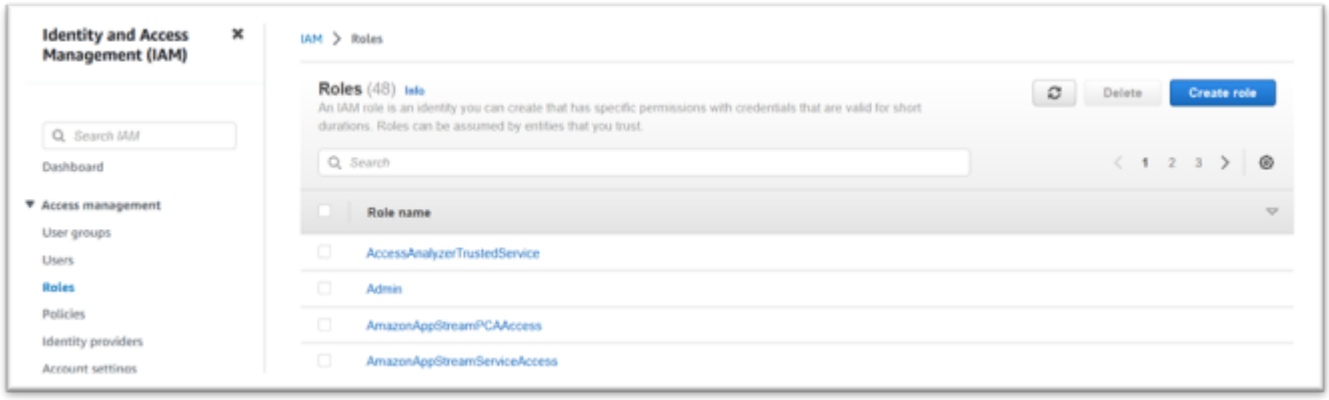


8. Scegli Crea policy.

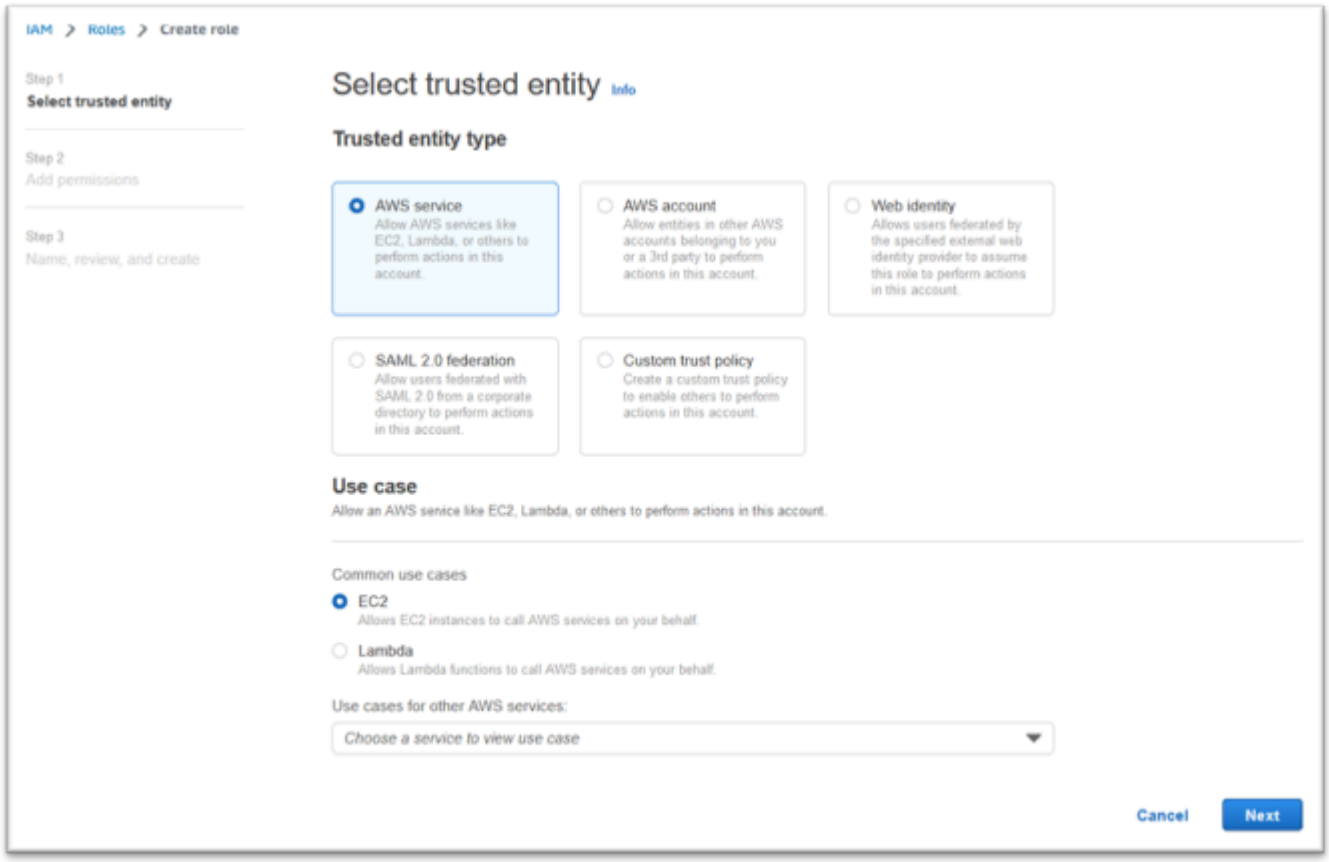
Crea il ruolo IAM

Dopo aver creato una policy IAM, crei un ruolo IAM e lo colleghi alla policy.

1. Passa a IAM in Console di gestione AWS.
2. Scegli Ruoli e poi Crea ruolo.

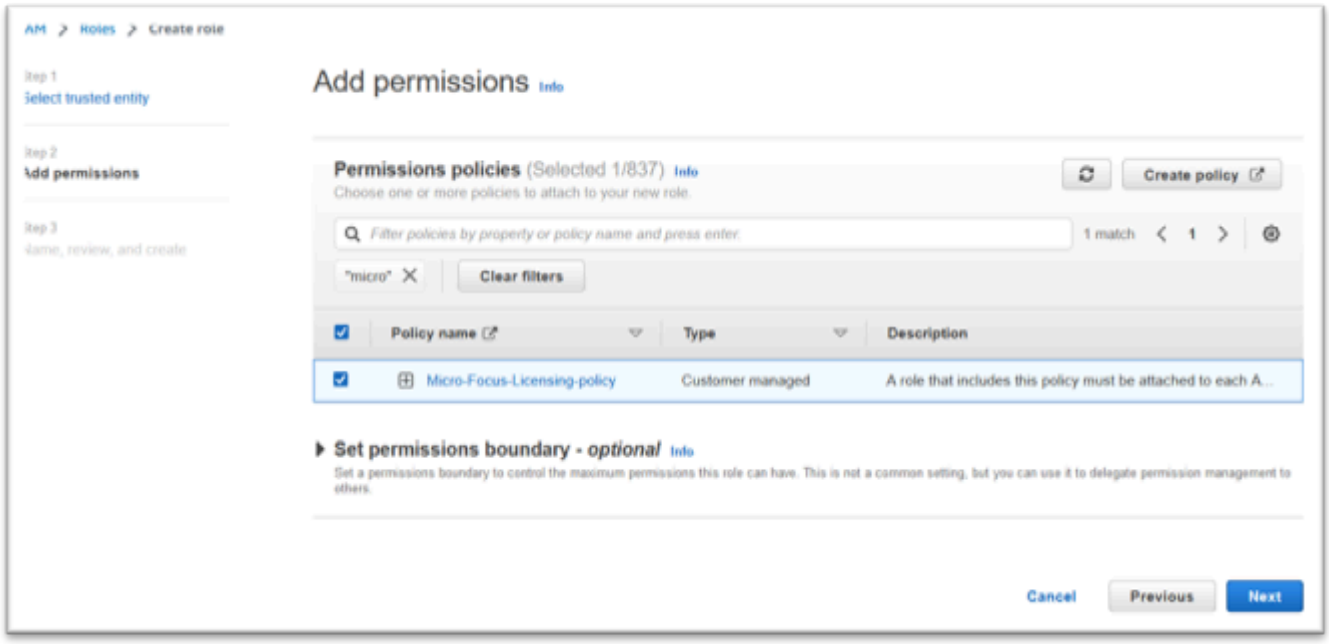


3. Lascia il tipo di entità affidabile come AWS servizio e scegli il caso d'uso comune di EC2.

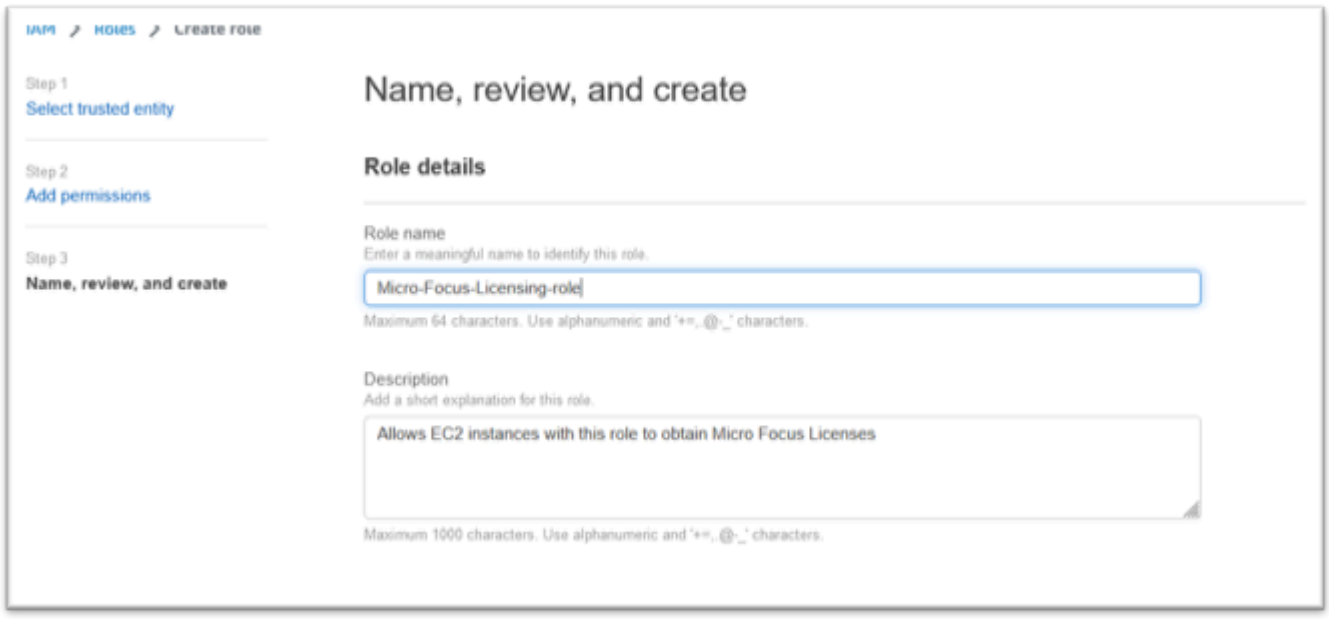


4. Scegli Next (Successivo).

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. Scegli Next (Successivo).



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».



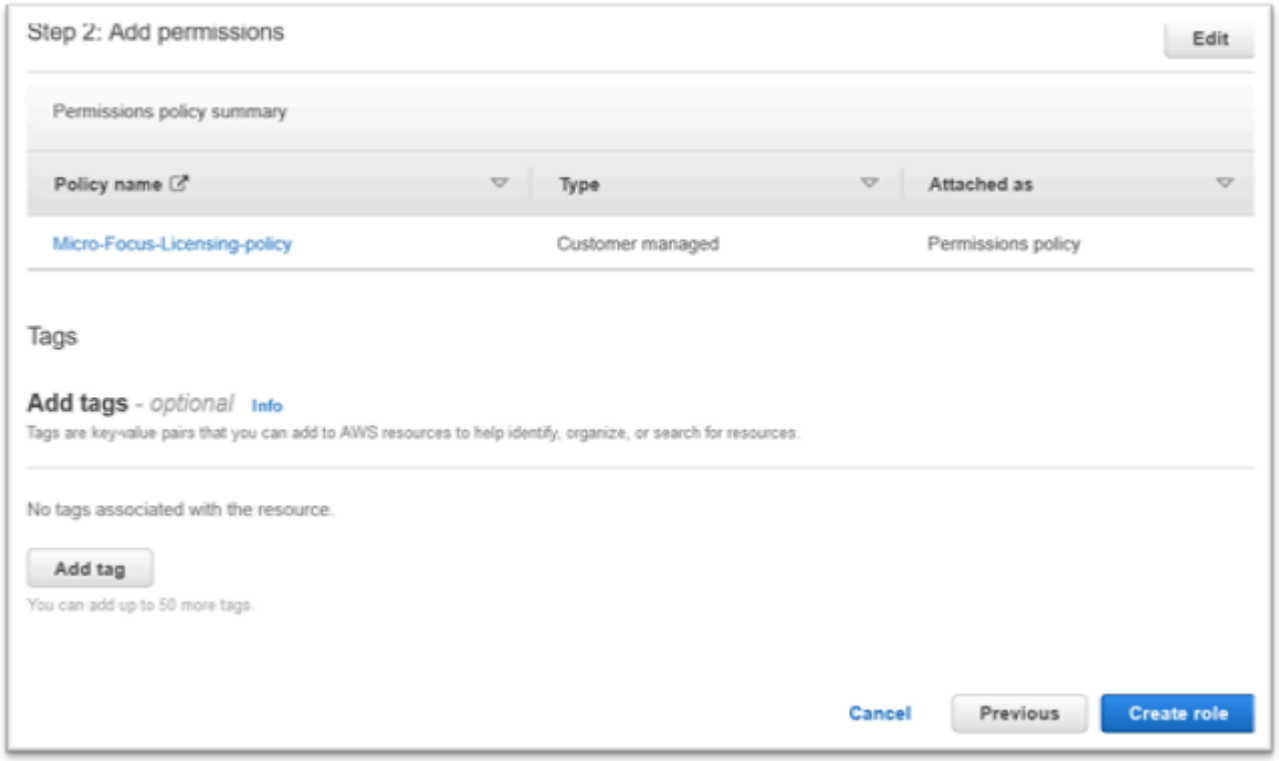
10. Nella fase 1: Seleziona entità attendibili, esamina il codice JSON e conferma che abbia i seguenti valori:

JSON

```
{
    "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.



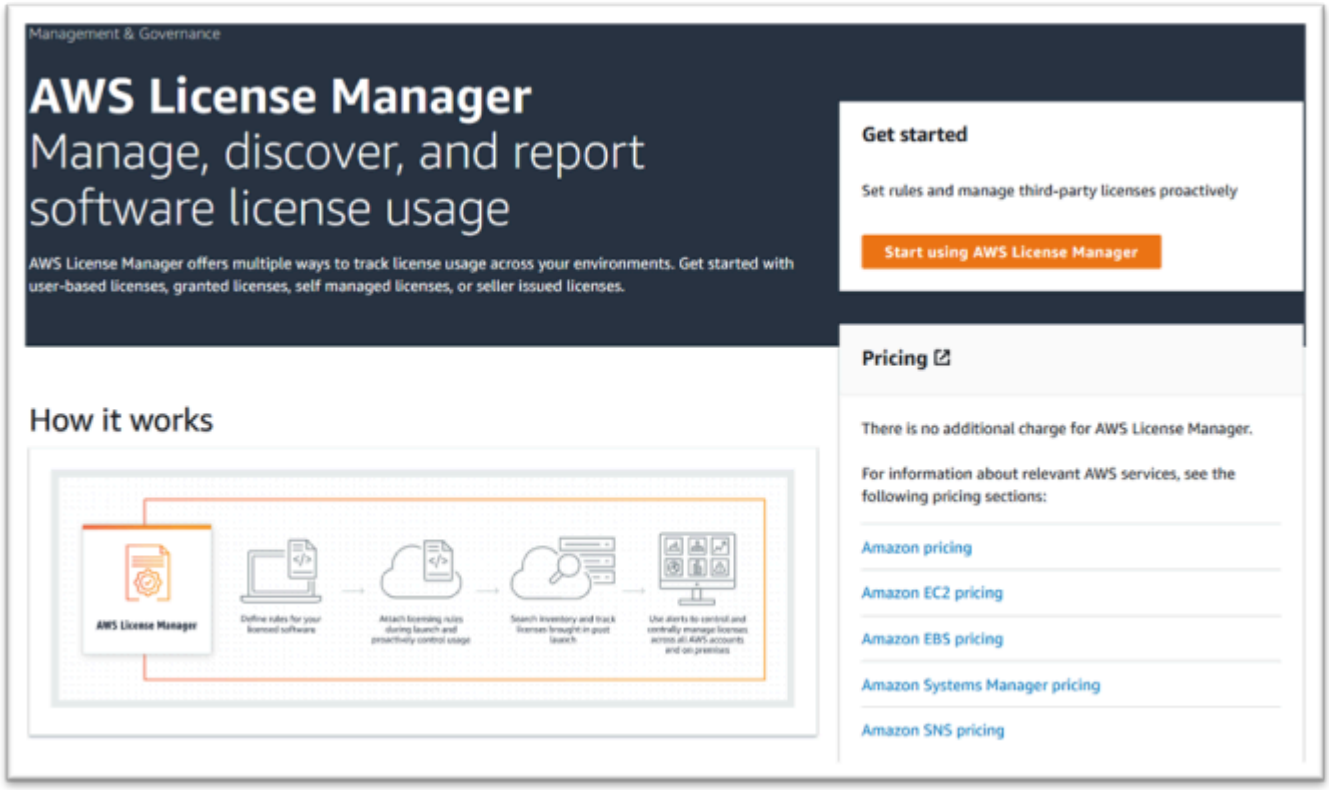
12. Scegli Crea ruolo.

Una volta completata la richiesta della lista consentita, continua con i seguenti passaggi.

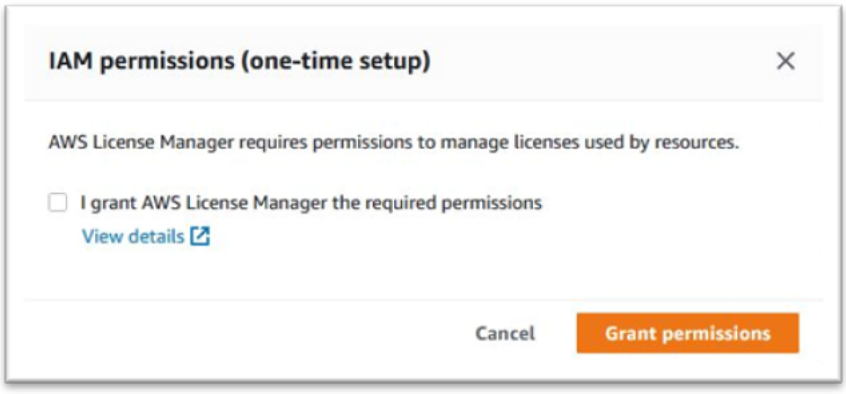
Concedi a License Manager le autorizzazioni richieste

Devi concedere le autorizzazioni AWS License Manager per configurare il motore di runtime Rocket Software (su Amazon EC2).

1. Passa a. AWS License Manager Console di gestione AWS



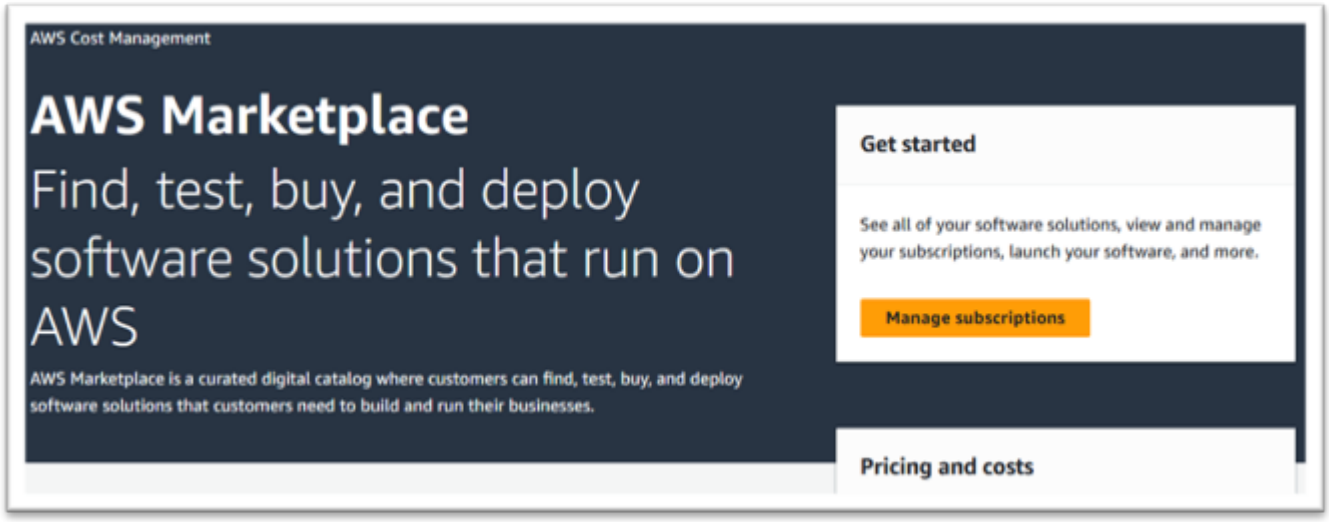
2. Scegli Inizia a usare AWS License Manager.
3. Se vedi il seguente pop-up, visualizza i dettagli, quindi seleziona la casella di controllo e premi Concedi autorizzazioni.



Iscriviti ad Amazon Machine Images

Dopo esserti abbonato a un Marketplace AWS prodotto, puoi avviare un'istanza dall'AMI del prodotto. Puoi anche gestire le AMI sottoscritte durante la configurazione del motore di runtime di Rocket Software (precedentemente Micro Focus) (su Amazon EC2).

1. Accedi a Abbonamenti in. Marketplace AWS Console di gestione AWS
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).



3. Copia e incolla uno dei seguenti link nella barra degli indirizzi del browser.

> **Note**
>
> 1. Scegli un link solo per uno dei prodotti che sei stato autorizzato a utilizzare.
> 2. Assicurati che il tuo account sia nella lista consentita seguendo la Richiedi l'aggiornamento della lista consentita per l'account pagina per utilizzare questi link.

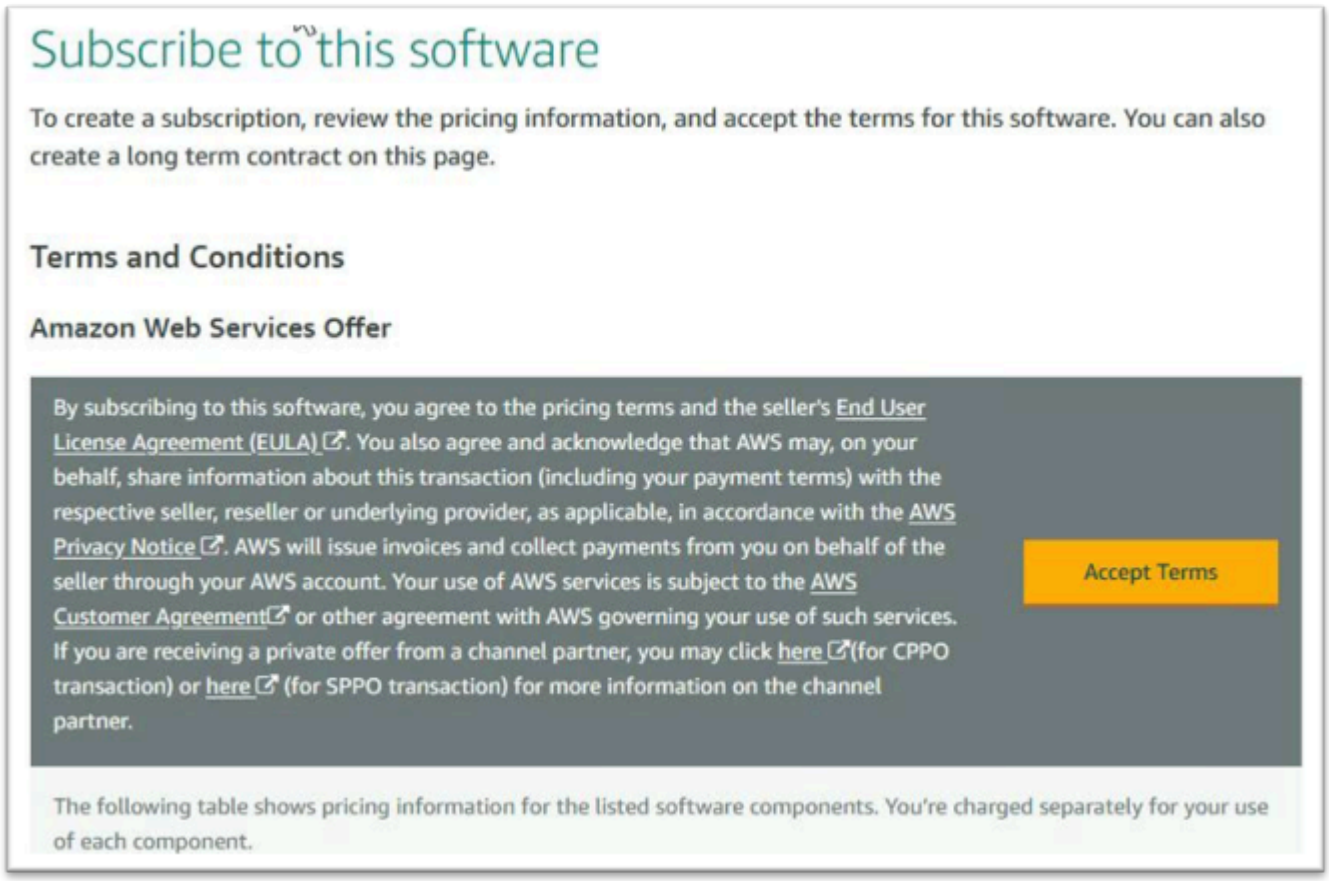
- Server aziendale: https://aws.amazon.com/marketplace/ pp/prodview-g5emev63l7blc
- Enterprise Server per Windows: pp/prodview-lwybsiyikbhc2 https://aws.amazon.com/marketplace/
- Sviluppatore aziendale: pp/prodview-77qmpr42yzxwk https://aws.amazon.com/marketplace/
- Sviluppatore aziendale con Visual Studio 2022: pp/prodview-m4l3lqiszo6cm https://aws.amazon.com/marketplace/
- Analizzatore aziendale: pp/prodview-tttheylcmcihm https://aws.amazon.com/marketplace/
- Strumenti Enterprise Build per Windows: pp/prodview-2rw35bbt6uozi https://aws.amazon.com/marketplace/
- Enterprise Stored Procedure https://aws.amazon.com/marketplace/: pp/prodview-zoeyqnsdsj6ha

- Stored Procedure aziendali con SQL Server 2019: pp/prodview-ynfklquwubnz4 https://aws.amazon.com/marketplace/

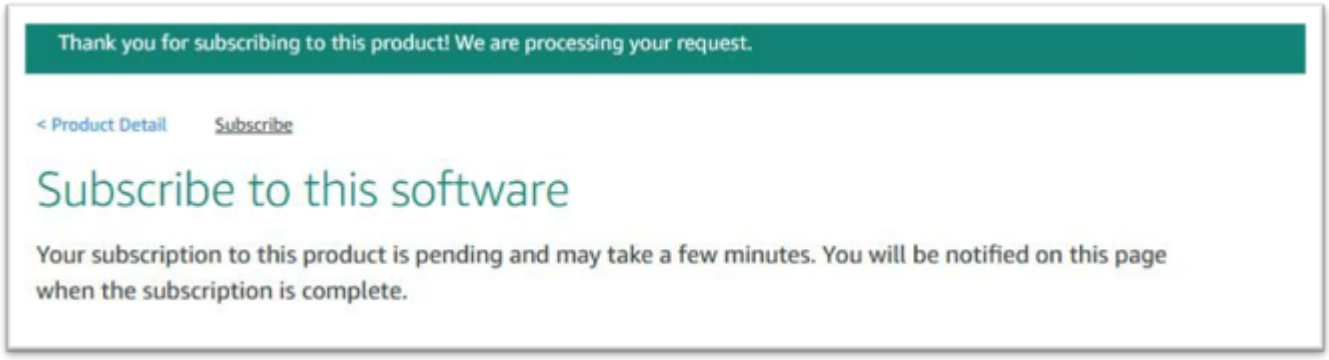
4. Scegliere Continue to Subscribe (Continua la sottoscrizione).



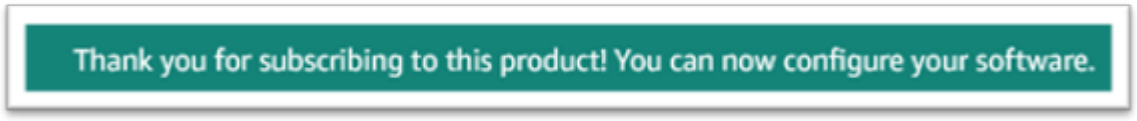
5. Se i Termini e condizioni sono accettabili, scegli Accetta termini.



6. L'elaborazione dell'abbonamento potrebbe richiedere alcuni minuti.

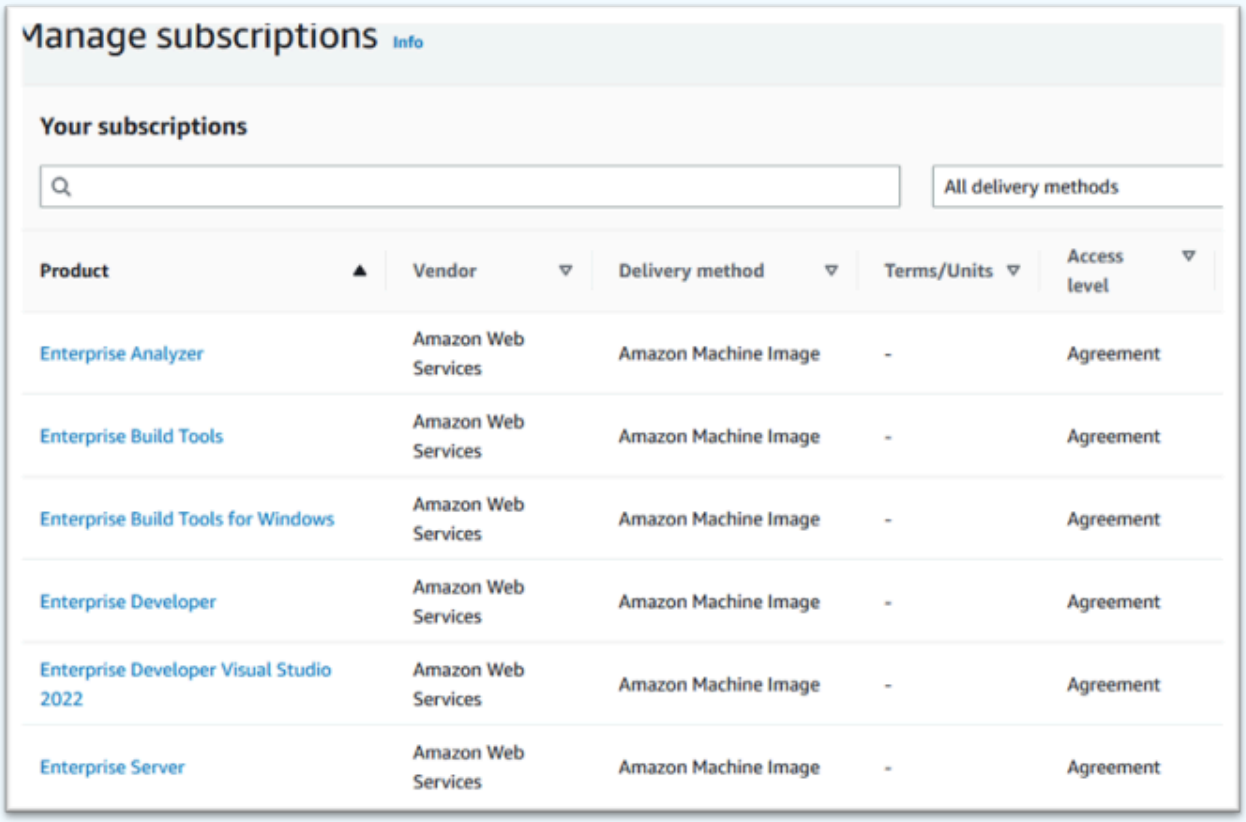


7. Dopo la visualizzazione del messaggio di ringraziamento, copia e incolla il link successivo dal passaggio 3 per continuare ad aggiungere abbonamenti.



8. Interrompi quando in Gestisci abbonamenti vengono visualizzati tutti gli abbonati AMIs.

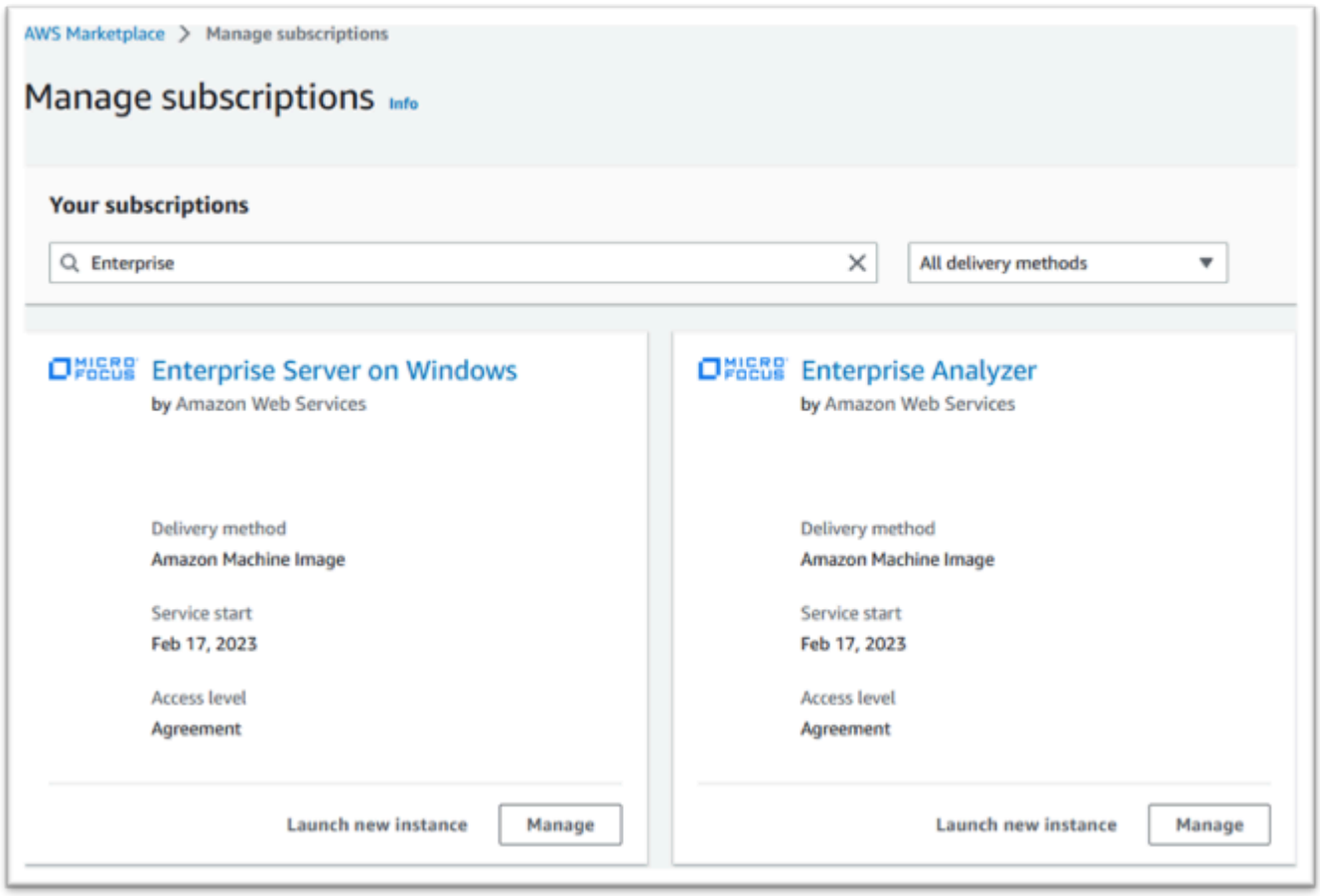
> **Note**
>
> Le preferenze del pannello (icona a forma di ingranaggio) sono impostate per mostrare la vista come tabella.



Avvia un'istanza di Modernizzazione del mainframe AWS Rocket Software (precedentemente Micro Focus)

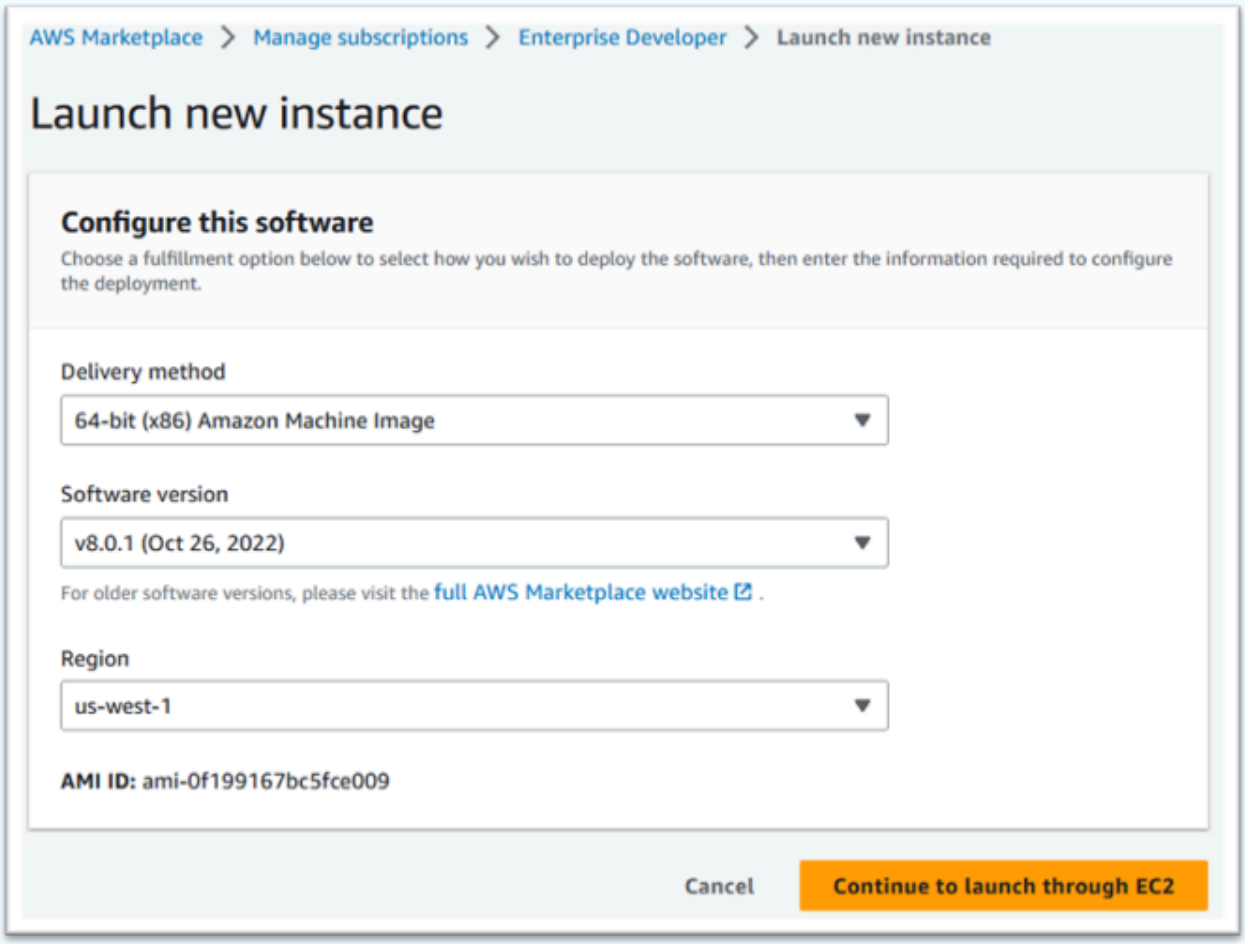
Dopo aver creato gli endpoint, la policy IAM, il ruolo IAM e aver sottoscritto AMIs, sei pronto per lanciare un'istanza di Modernizzazione del mainframe AWS Rocket Software (Micro Focus) in. Console di gestione AWS

1. Vai a Marketplace AWS Abbonamenti in. Console di gestione AWS
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 tutti Modernizzazione del mainframe AWS AMIs.

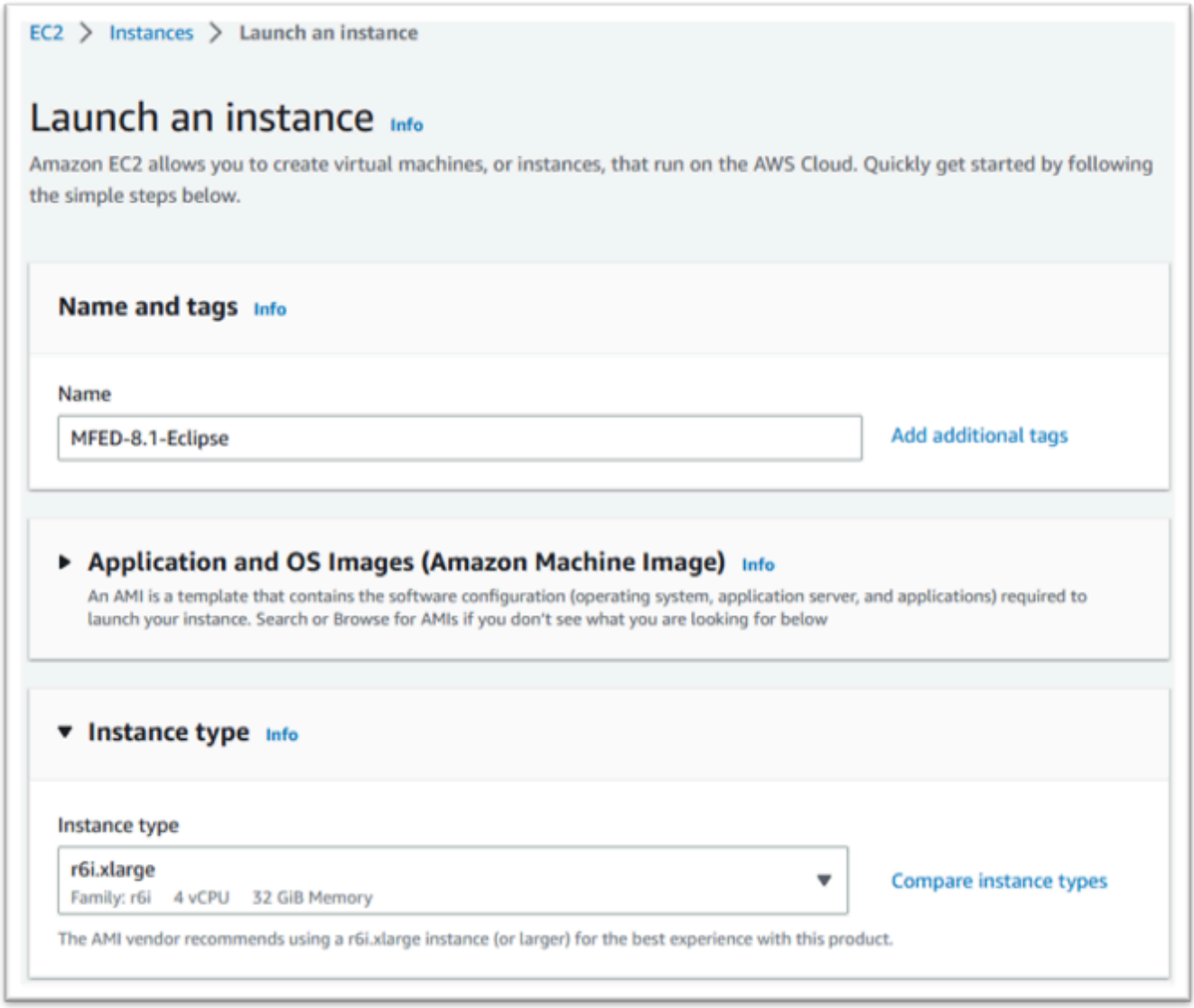


5. Inserisci un nome per il server.
6. Scegliere un tipo di istanza.

   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 Rocket Software Performance Availability Cluster (PAC) con scale-out, un r6i.large

   > **Note**
   >
   > La sezione Immagini dell'applicazione e del sistema operativo è stata compressa per quanto riguarda la schermata.



7. Scegliete o create (e salvate) una coppia di chiavi (non mostrata).

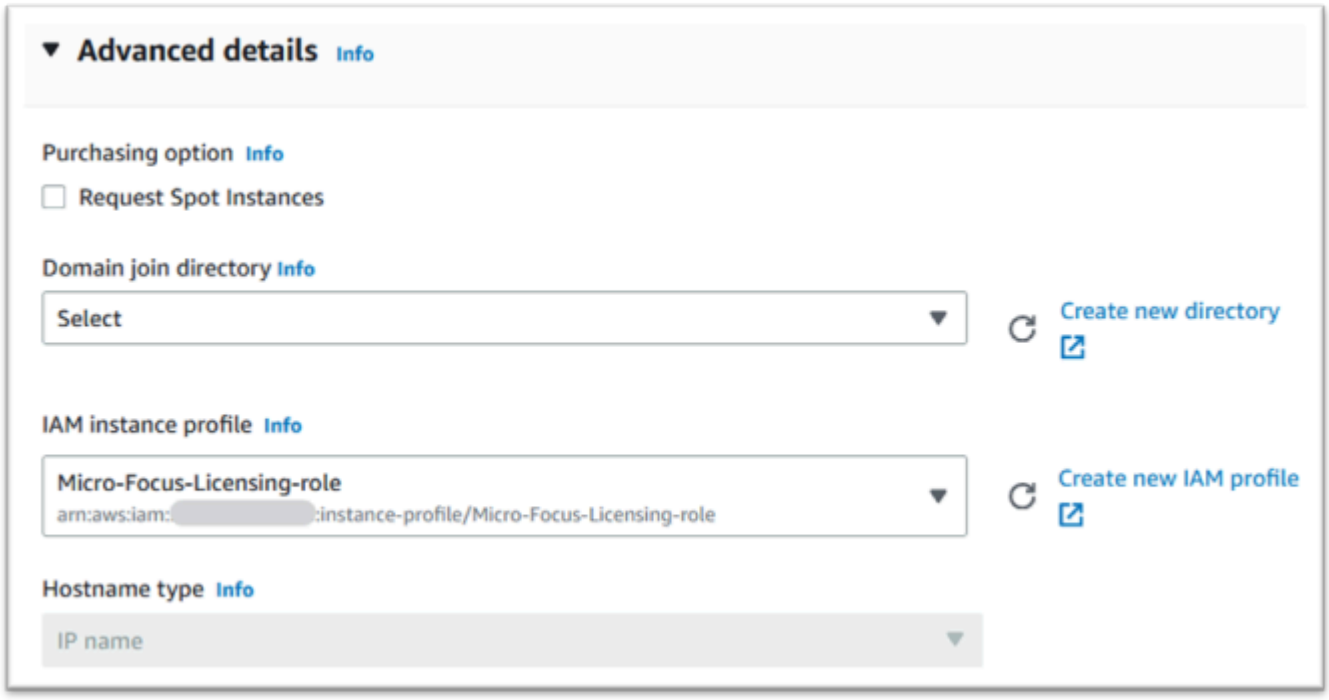
   Per ulteriori informazioni sulle coppie di chiavi per le istanze Linux, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

   Per ulteriori informazioni sulle coppie di chiavi per le istanze Windows, consulta Coppie di chiavi Amazon EC2 e istanze Windows.

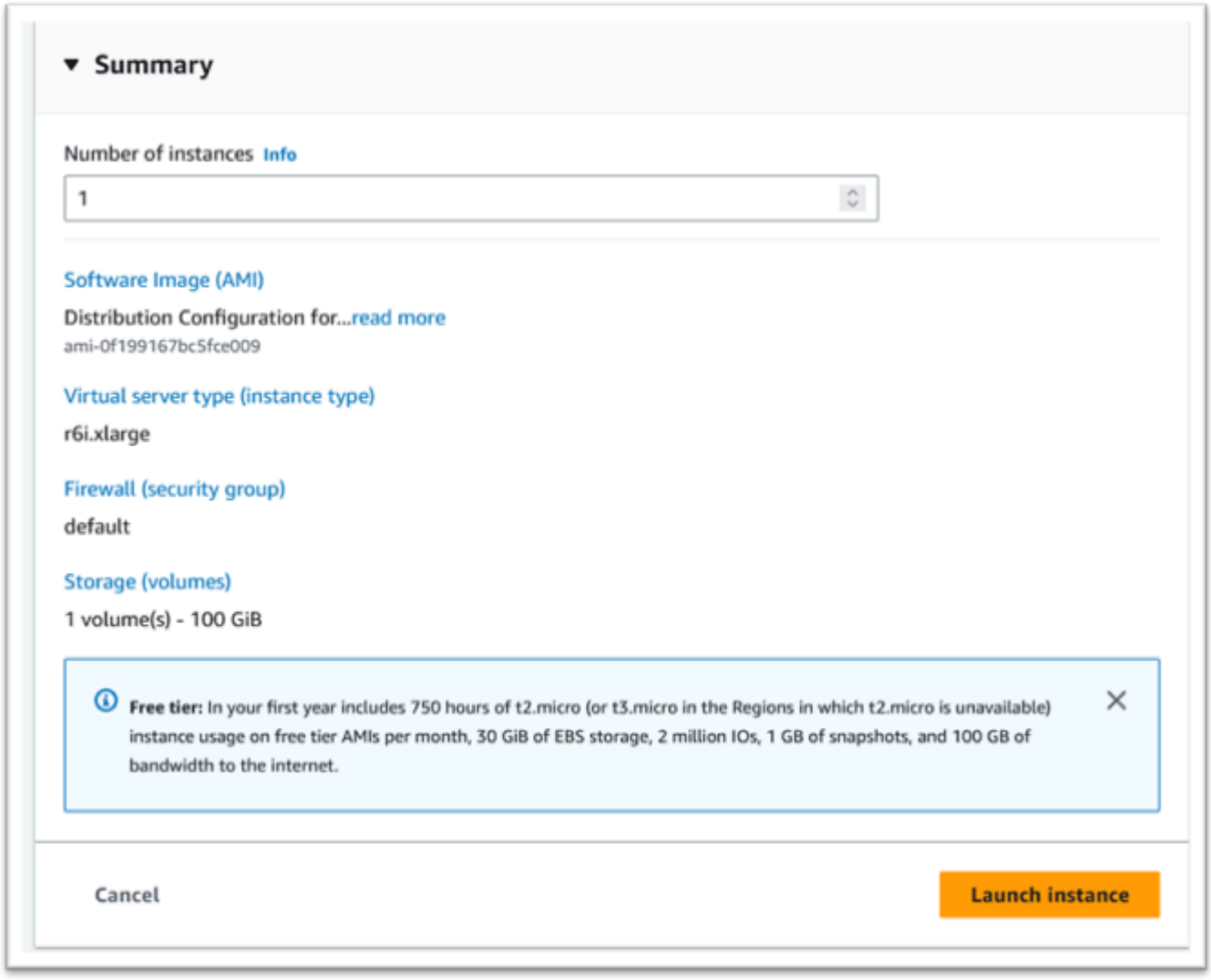
8. Modifica le impostazioni di rete e scegli il VPC nell'elenco consentito e la sottorete appropriata.
9. Scegli o crea un gruppo di sicurezza. Se si tratta di un'istanza EC2 di Enterprise Server, è normale consentire al traffico TCP verso le porte 86 e 10086 per amministrare la configurazione del software Rocket.

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.

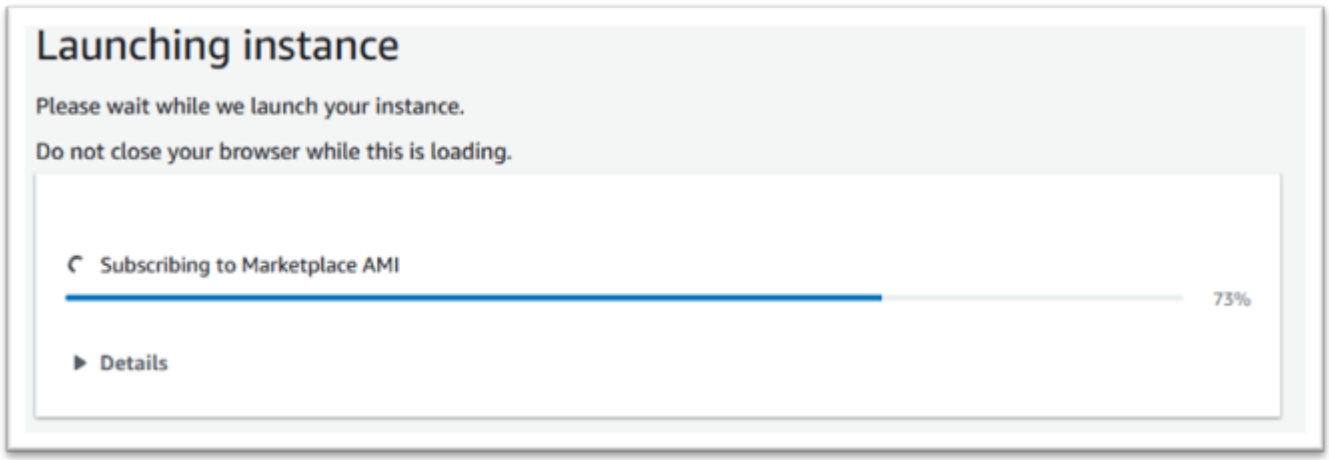


12. Esamina il riepilogo e invia Launch Instance.

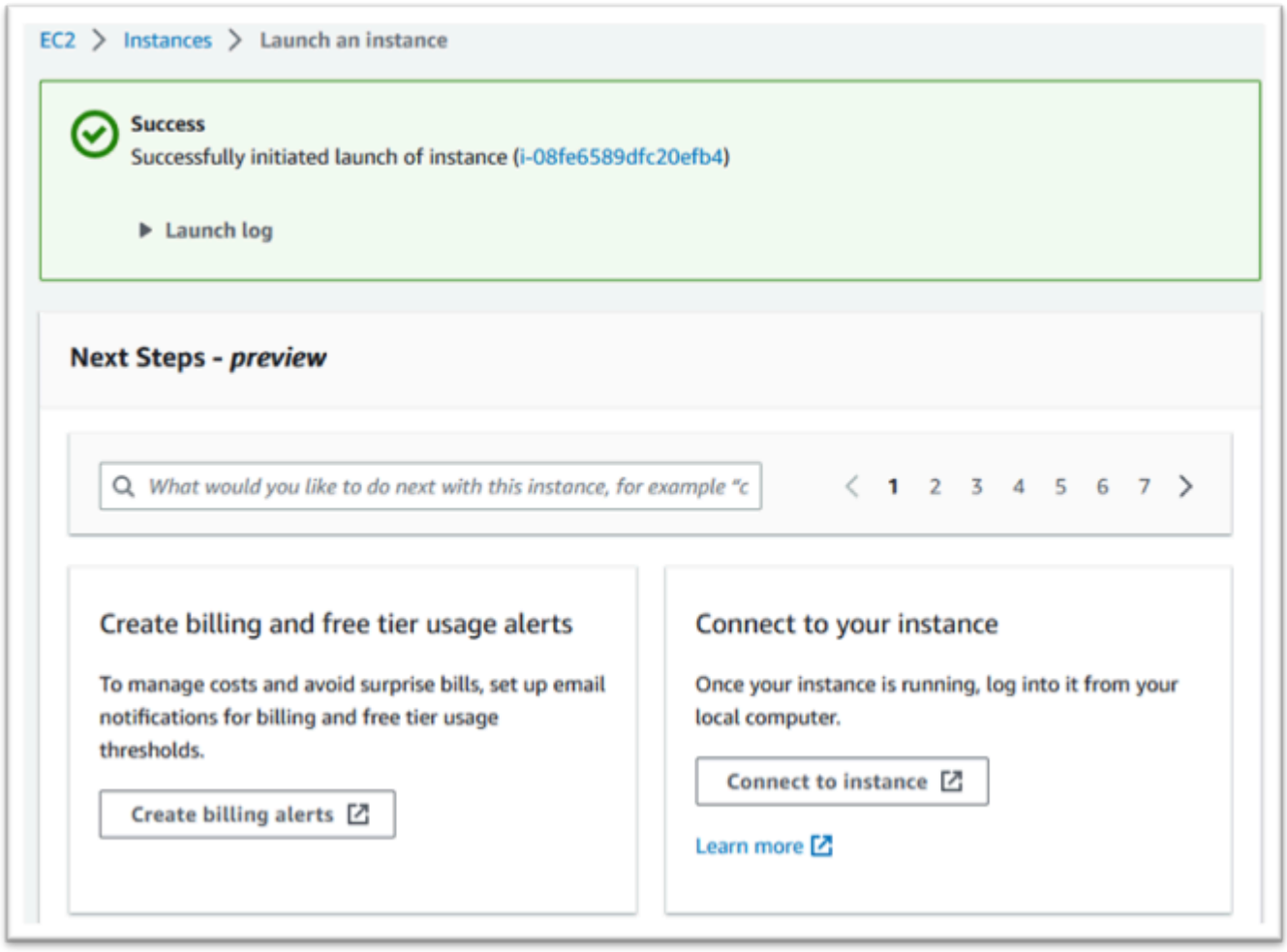


13. L'avvio dell'istanza avrà esito negativo se viene scelto un tipo di server virtuale non valido.

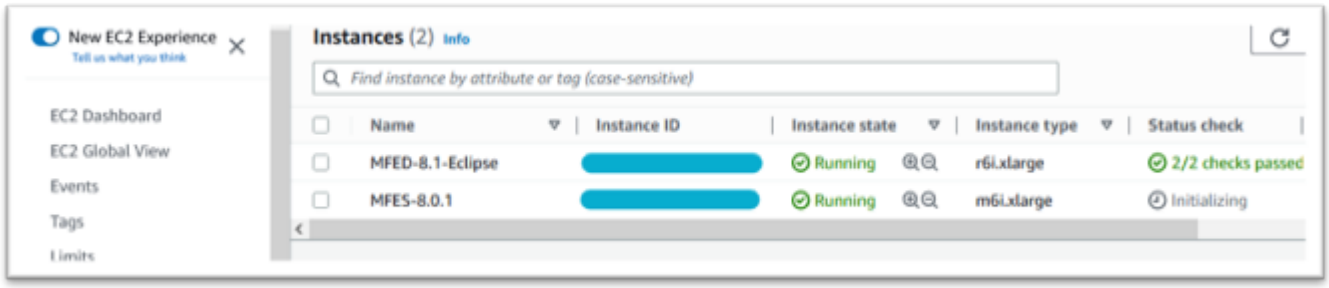
    In tal caso, scegli Modifica configurazione dell'istanza e modifica il tipo di istanza.



14. Una volta visualizzato il messaggio «Success», scegli Connetti all'istanza per ottenere i dettagli della connessione.



15. In alternativa, accedi a EC2 nel Console di gestione AWS.
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*.gestore delle licenze
- com.amazonaws. *region*.sts

I passaggi precedenti hanno definito com.amazonaws. *region*Il servizio.s3 come endpoint gateway. Questo endpoint necessita di una voce nella tabella di routing per tutte le sottoreti senza accesso a Internet.

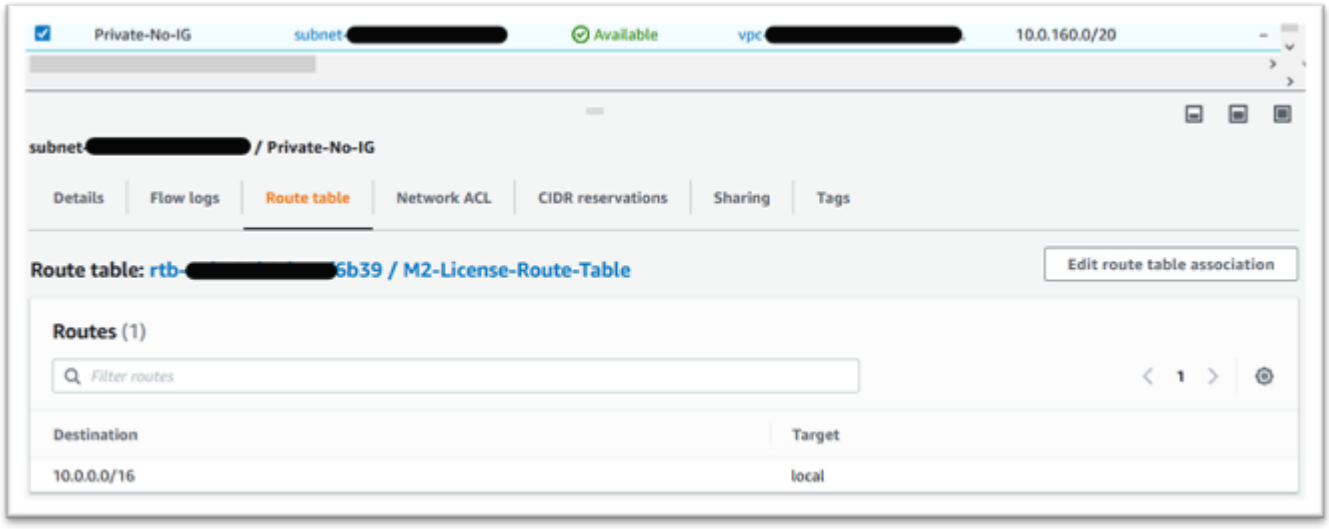
I tre servizi aggiuntivi verranno definiti come endpoint di interfaccia.

Argomenti

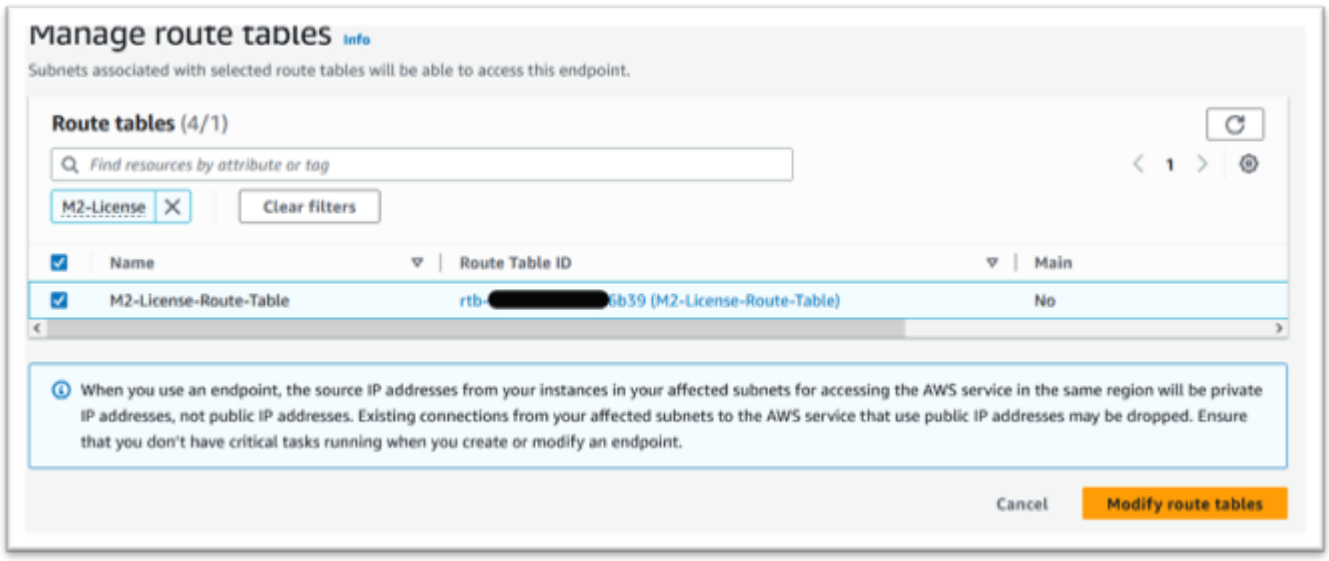
- Aggiungi la voce della tabella Route per l'endpoint Amazon S3
- Definire il gruppo di sicurezza richiesto
- Crea gli endpoint del servizio

Aggiungi la voce della tabella Route per l'endpoint Amazon S3

1. Vai a VPC in Console di gestione AWS 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.

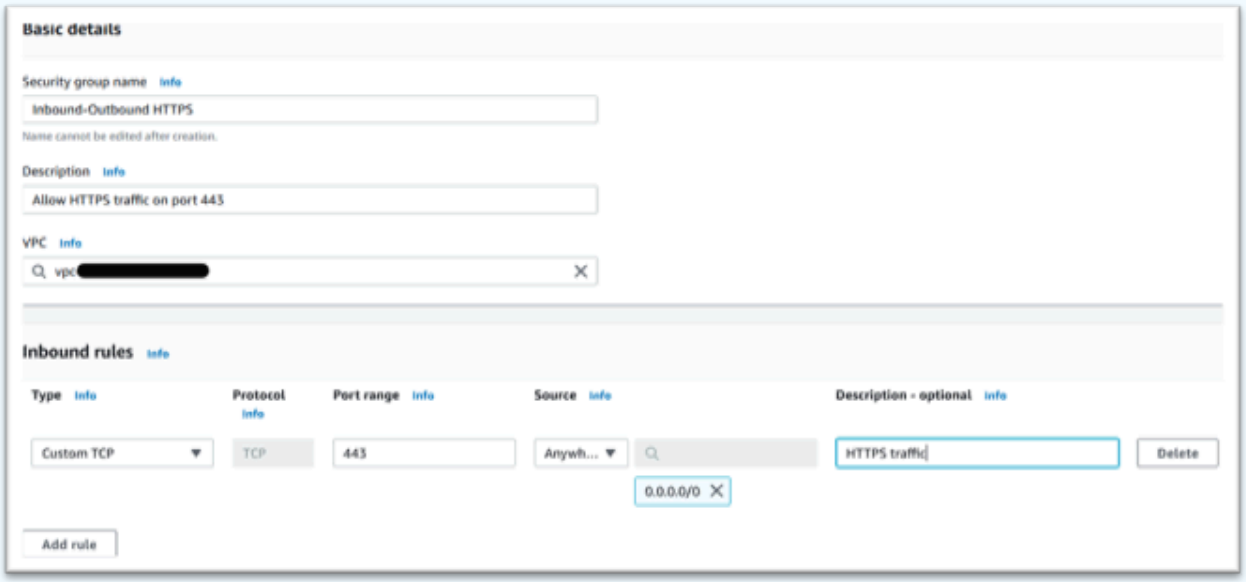


Definire il gruppo di sicurezza richiesto

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. Console di gestione AWS
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.



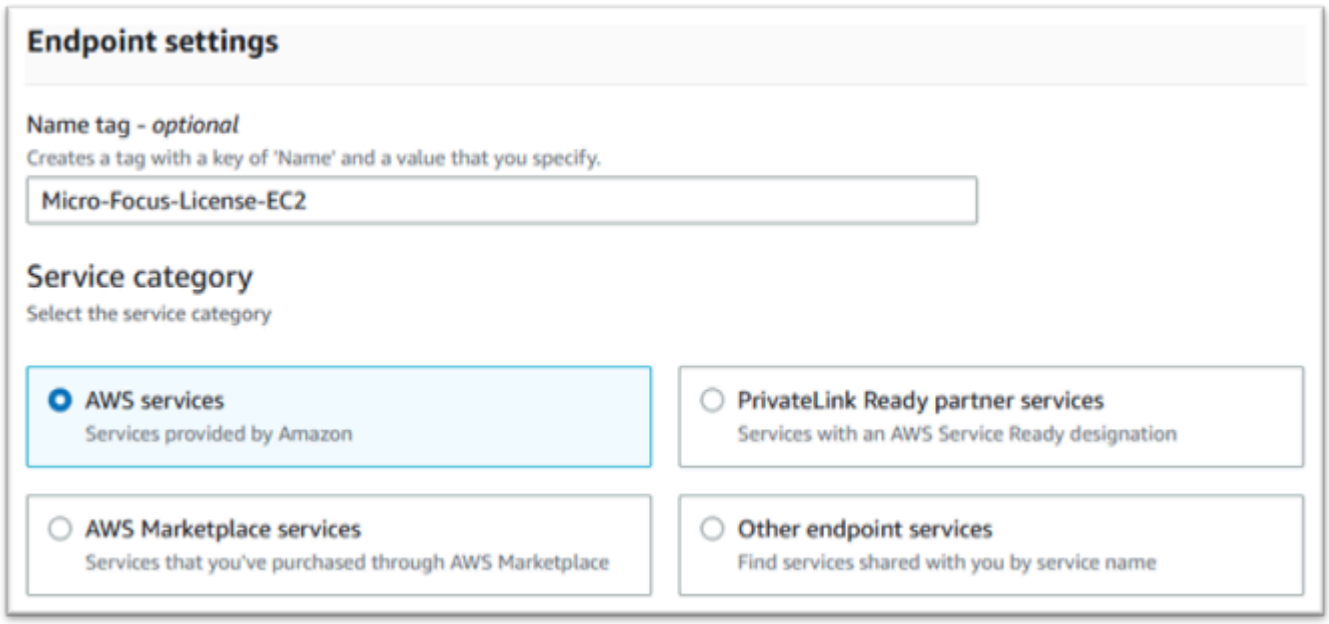
6. Premi Crea gruppo di sicurezza.

Crea gli endpoint del servizio

Ripeti questo processo tre volte, una volta per ogni servizio.

1. Accedi ad Amazon VPC in Console di gestione AWS 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.



5. In Servizi cerca il servizio di interfaccia corrispondente, che è uno dei seguenti:

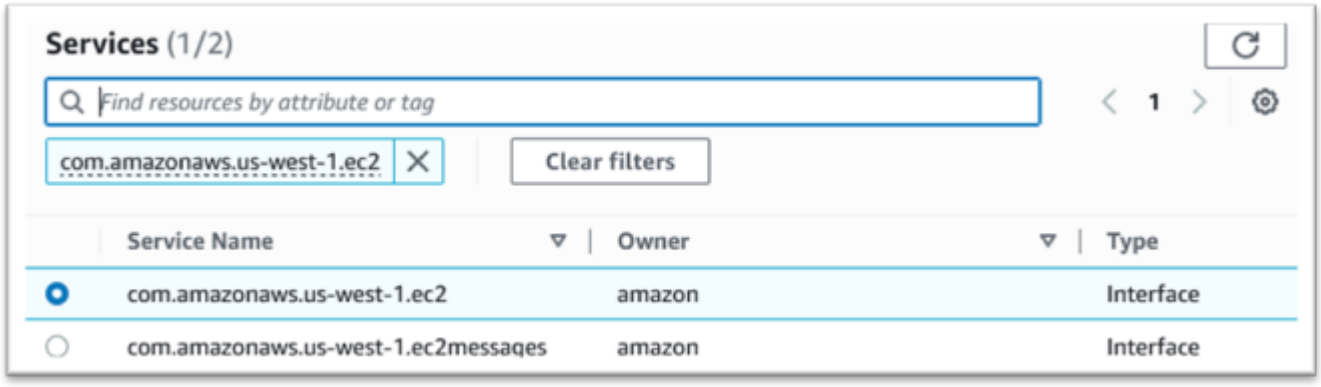
   - «com.amazonaws. *region*.ec2»
   - «.com» amazonaws. *region*.st»
   - «.com» amazonaws. *region*.gestore delle licenze»

   Esempio:

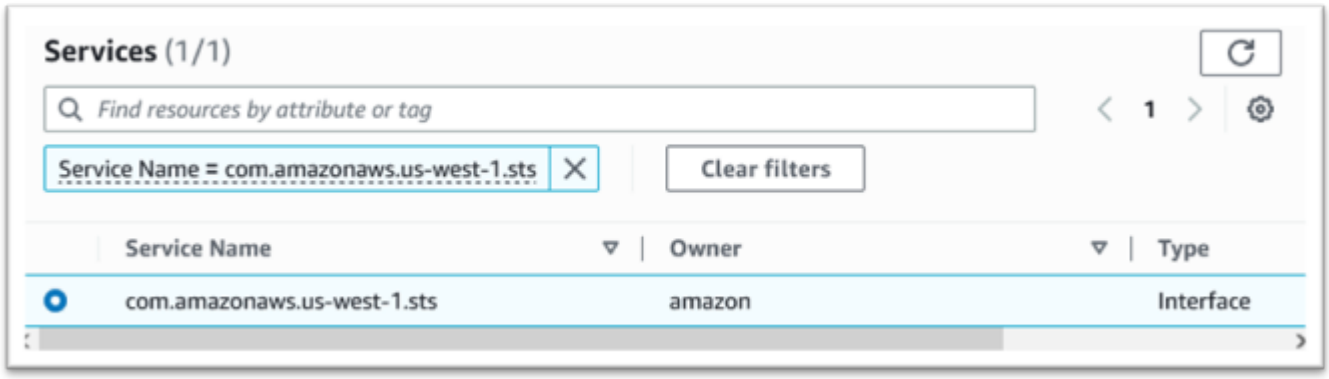
   - «com.amazonaws.us-west-1.ec2»
   - «com.amazonaws.us-west-1.sts»
   - «com.amazonaws.us-west-1.license-manager»

6. Scegli il servizio di interfaccia corrispondente.

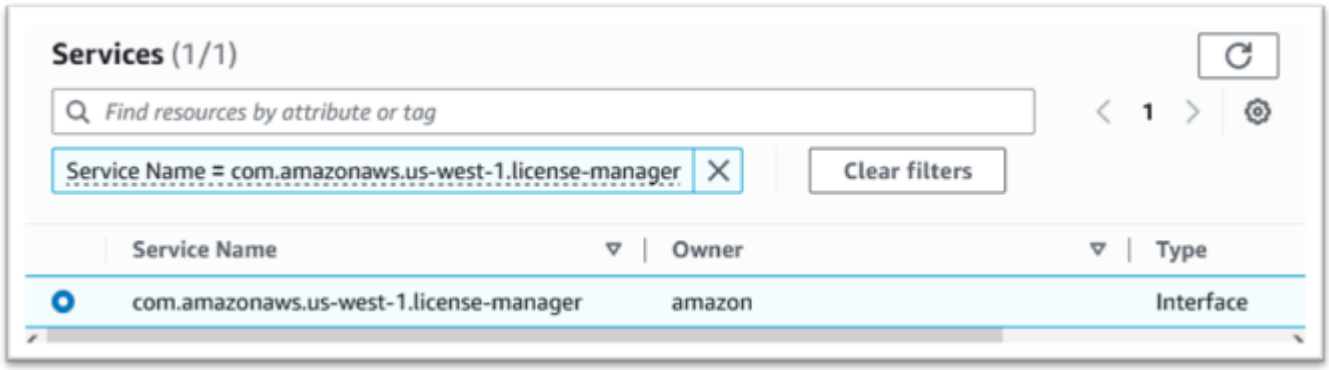
   com.amazonaws. ***region***.ec2:



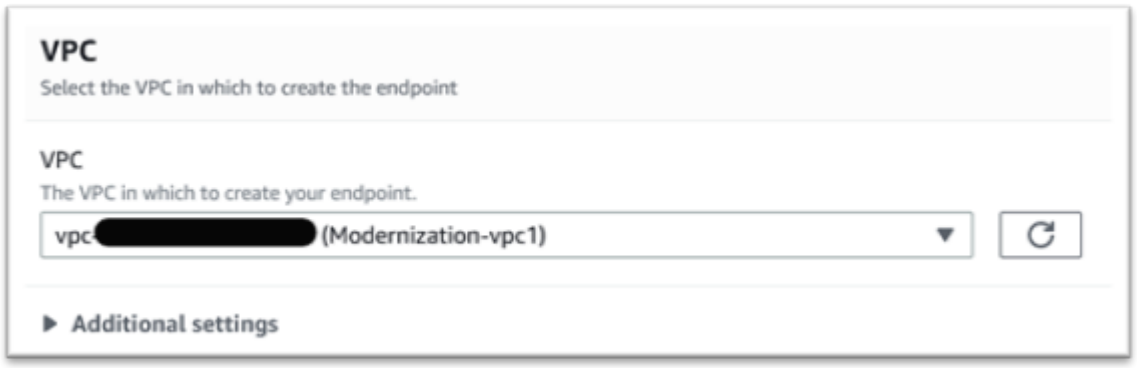
com.amazonaws. ***region***.st:



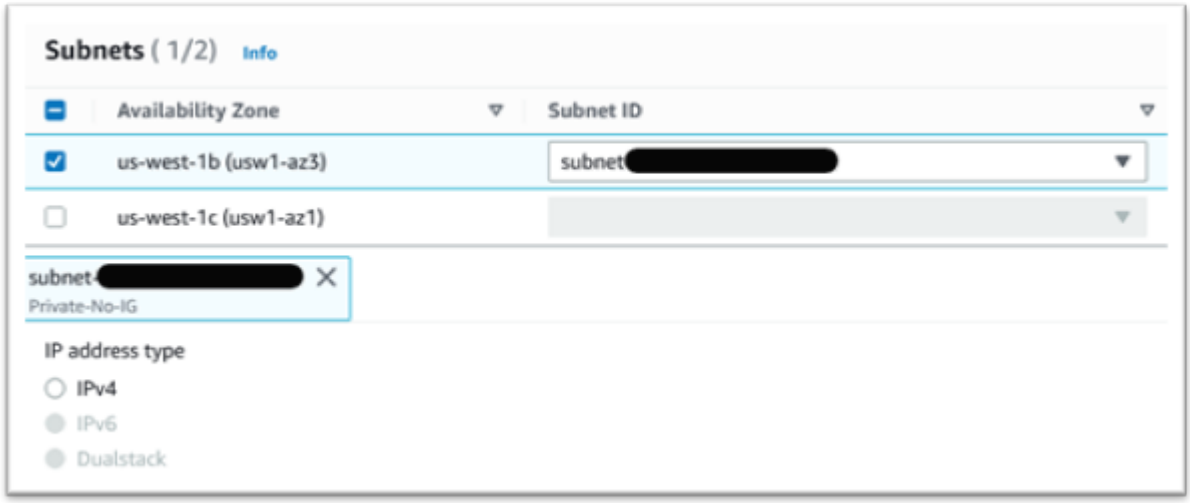
com.amazonaws. ***region***.gestore delle licenze:



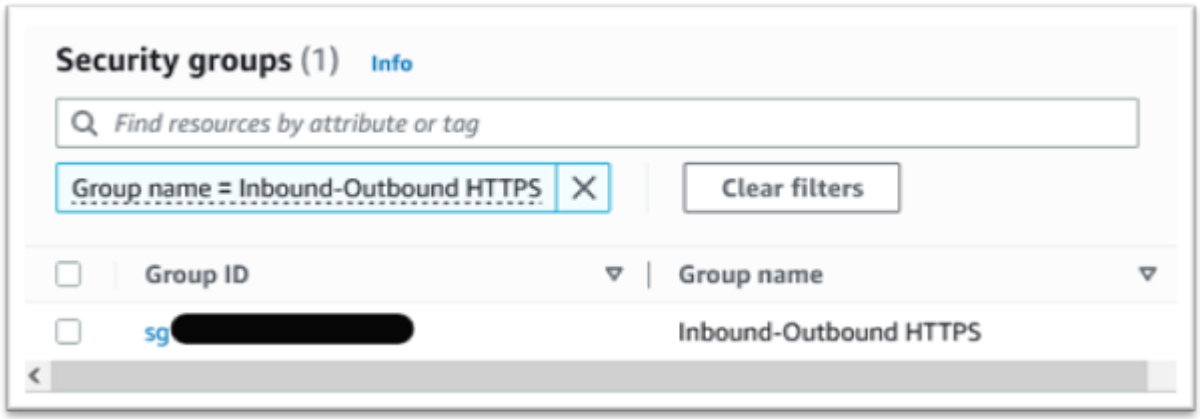
7. Per VPC scegli il VPC per l'istanza.



8. Scegli la zona di disponibilità e le sottoreti per il VPC.



9. Scegli il gruppo di sicurezza creato in precedenza.



10. In Policy scegli Accesso completo.



11. Scegliere Create Endpoint (Crea endpoint).
12. Ripetere questa procedura per le interfacce rimanenti.

Configurazione dell'automazione per le sessioni di streaming di Rocket Enterprise Analyzer (precedentemente Micro Focus) e Rocket 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 WorkSpaces delle applicazioni, consulta Use Session Scripts to Manage Your AppStream 2.0 Users' Streaming Experience nella Amazon WorkSpaces Applications Administration Guide.

Questa funzionalità richiede che siano disponibili almeno le seguenti versioni delle immagini di Enterprise Analyzer ed Enterprise Developer:

- `m2-enterprise-analyzer-v8.0.4.R1`
- `m2-enterprise-developer-v8.0.4.R1`

Argomenti

- Configurare l'automazione all'inizio della sessione
- Configura l'automazione al termine della sessione

Configurare l'automazione all'inizio della sessione

Se desideri eseguire uno script di automazione quando gli utenti si connettono alle WorkSpaces applicazioni, crea lo script e assegnagli un nome`m2-user-setup.cmd`. Archivia lo script nella cartella Home WorkSpaces delle applicazioni dell'utente. Le immagini WorkSpaces delle applicazioni 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 WorkSpaces Applications, che attualmente è di 60 secondi. Per ulteriori informazioni, consulta Run Scripts Before Streaming Sessions Begin nella Amazon WorkSpaces Applications Administration Guide.

Configura l'automazione al termine della sessione

Se desideri eseguire uno script di automazione quando gli utenti si disconnettono dalle WorkSpaces applicazioni, crea lo script e assegnagli un nome. `m2-user-teardown.cmd` Archivia lo script nella cartella Home WorkSpaces delle applicazioni dell'utente. Le immagini WorkSpaces delle applicazioni 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 WorkSpaces Applications, che attualmente è di 60 secondi. Per ulteriori informazioni, consulta Run Scripts After Streaming Sessions End nella Amazon WorkSpaces Applications Administration Guide.

Visualizza i set di dati come tabelle e colonne in Rocket Enterprise Developer (precedentemente Micro Focus Enterprise Developer)

È possibile accedere ai set di dati mainframe distribuiti in AWS Mainframe Modernization utilizzando il runtime del software Rocket (precedentemente Micro Focus). È possibile visualizzare i set di dati migrati come tabelle e colonne da un'istanza di Rocket 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
- Passaggio 1: configurare la connessione ODBC al datastore di Rocket Software (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 di Rocket Software (precedentemente Micro Focus) come tabelle e colonne

Prerequisiti

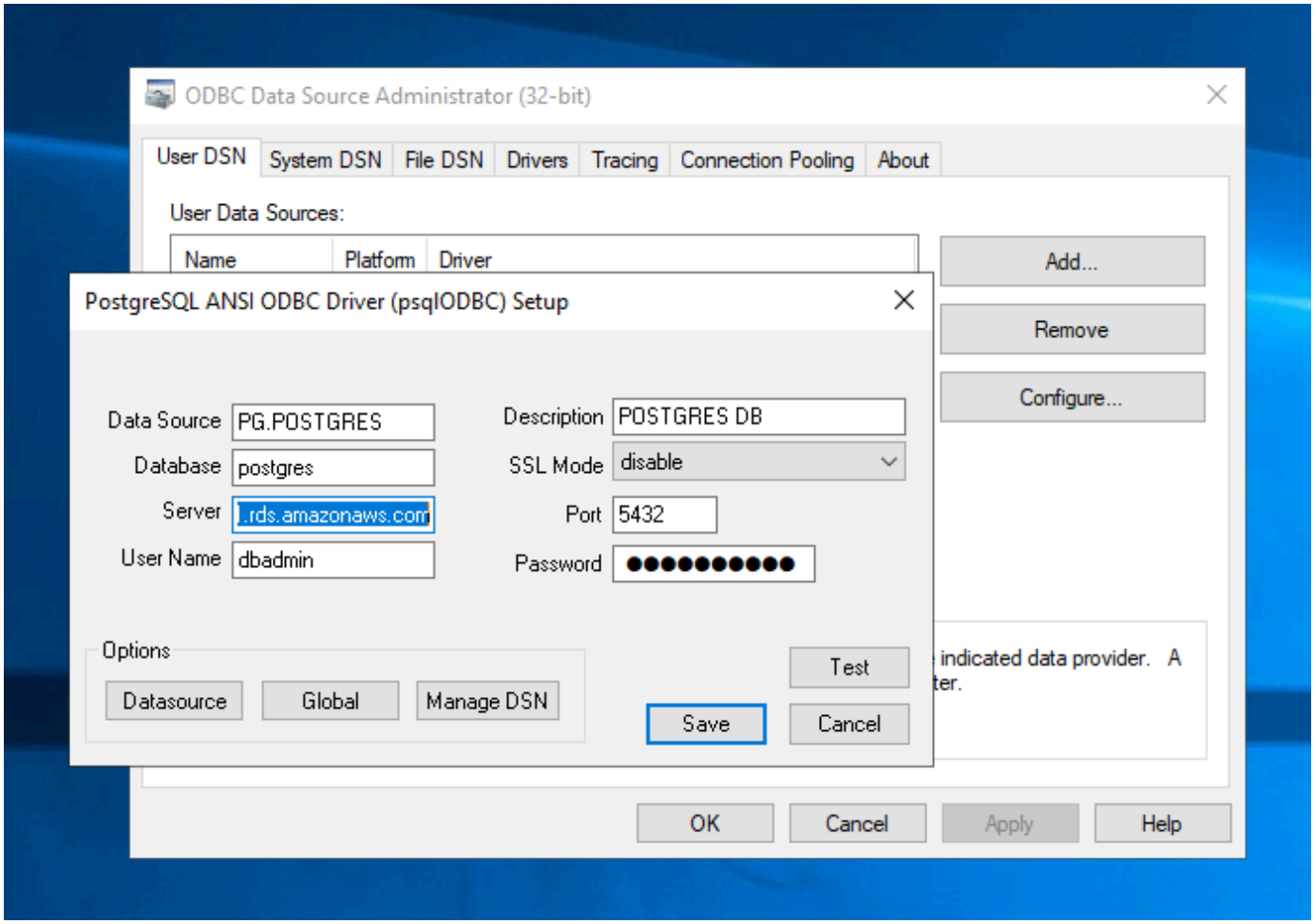
- È necessario avere accesso a Rocket Enterprise Developer Desktop tramite WorkSpaces Applicazioni.
- È necessario che un'applicazione sia distribuita e funzionante in AWS Mainframe Modernization utilizzando il motore di runtime Rocket Software.
- Stai archiviando i dati dell'applicazione in Aurora PostgreSQL Compatible Edition.

Passaggio 1: configurare la connessione ODBC al datastore di Rocket Software (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 che si effettua una sola volta.

1. Accedi a Rocket Enterprise Developer Desktop utilizzando l'URL di streaming WorkSpaces delle applicazioni.
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
```



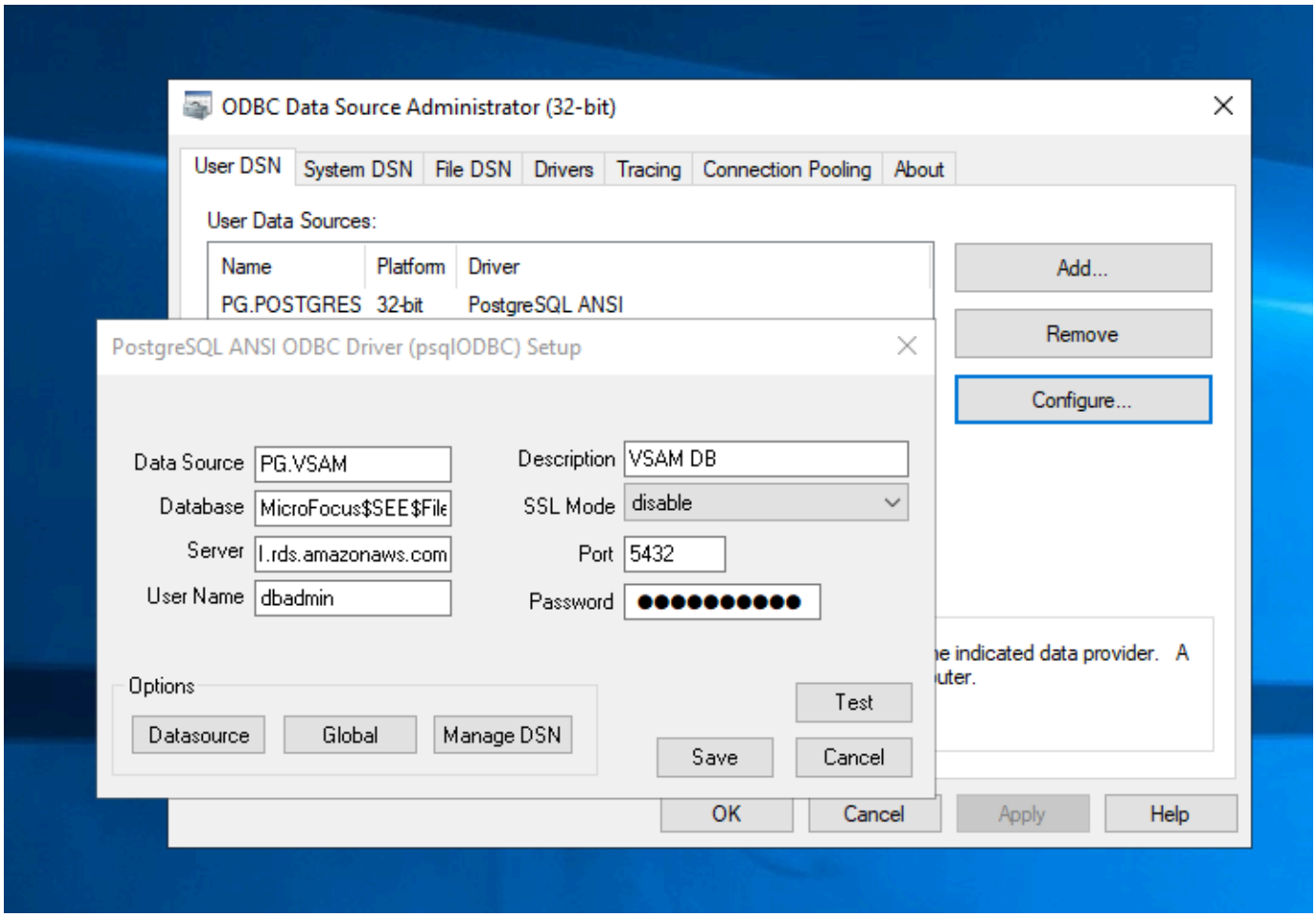
5. Scegli Test per assicurarti che la connessione funzioni. Dovresti vedere il messaggio `Connection successful` se il test ha esito positivo.

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 per`PG.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. Si tratta di un passaggio di configurazione che si effettua una sola volta.

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.
```

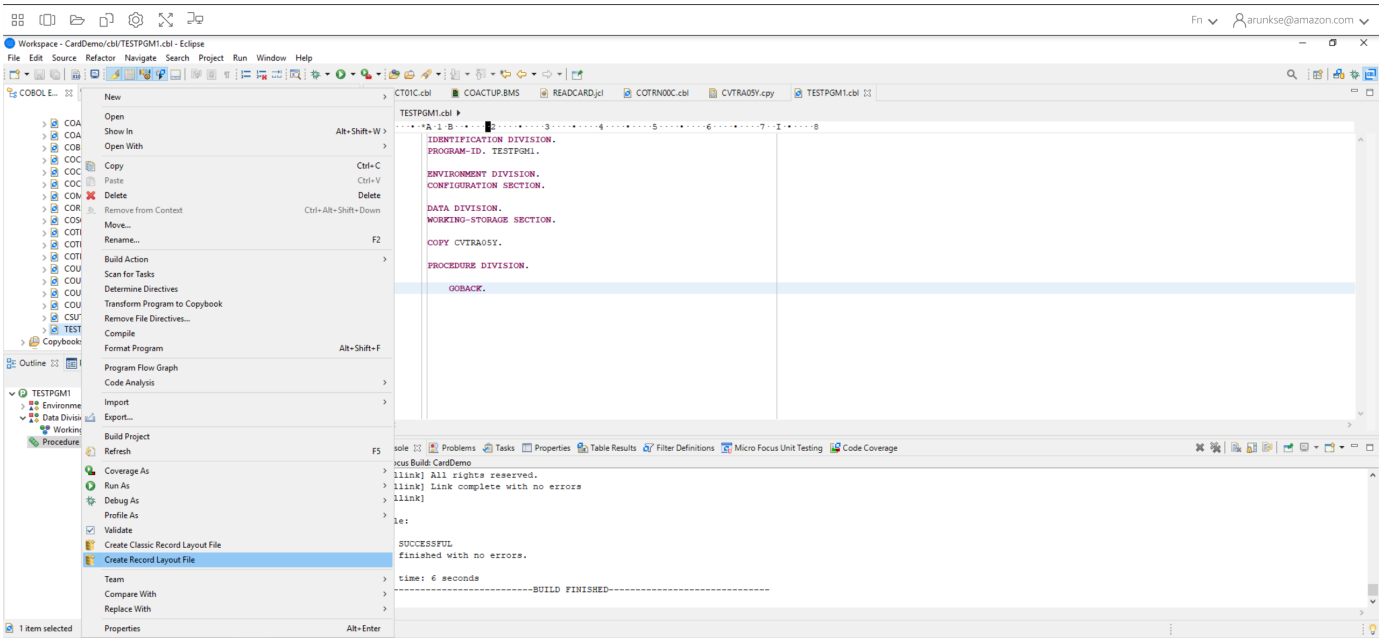
```
       DATA DIVISION.
       WORKING-STORAGE SECTION.

       COPY CVTRA05Y.

       PROCEDURE DIVISION.

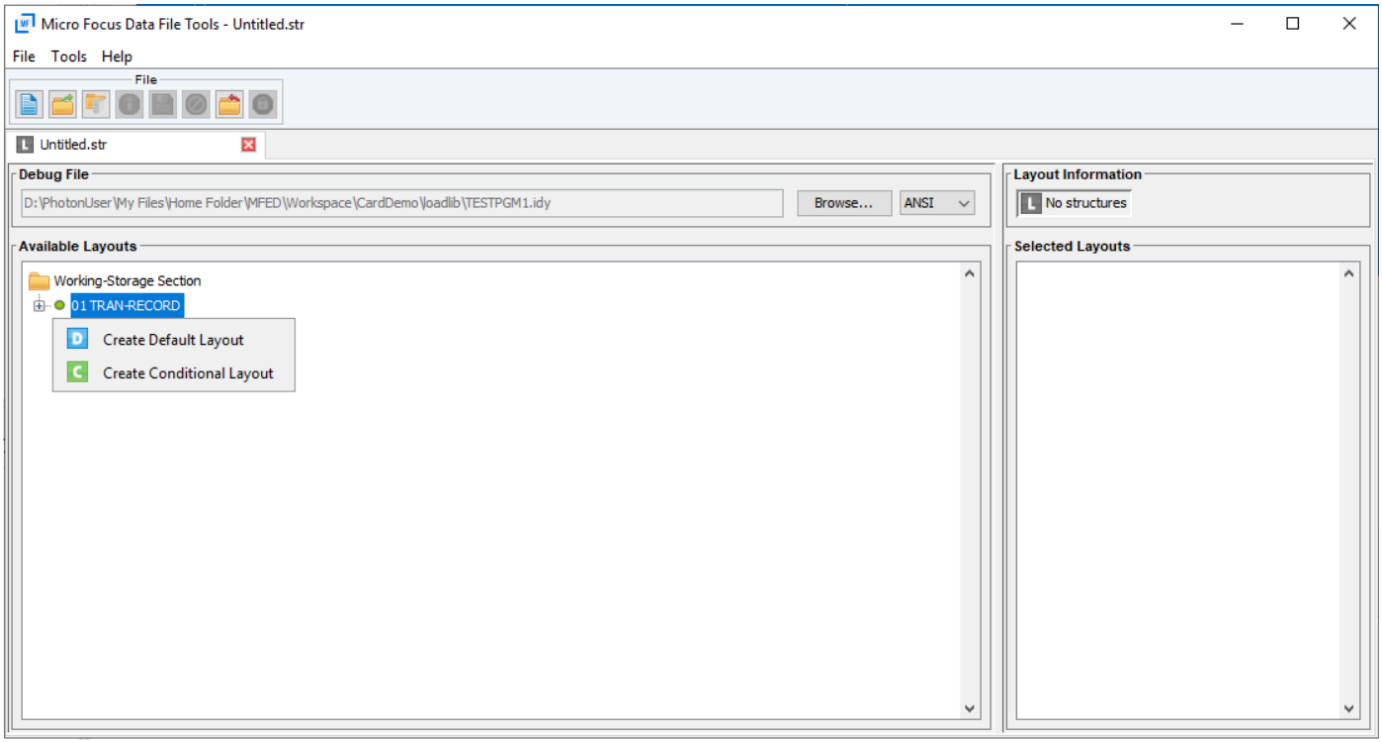
       GOBACK.
```

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.

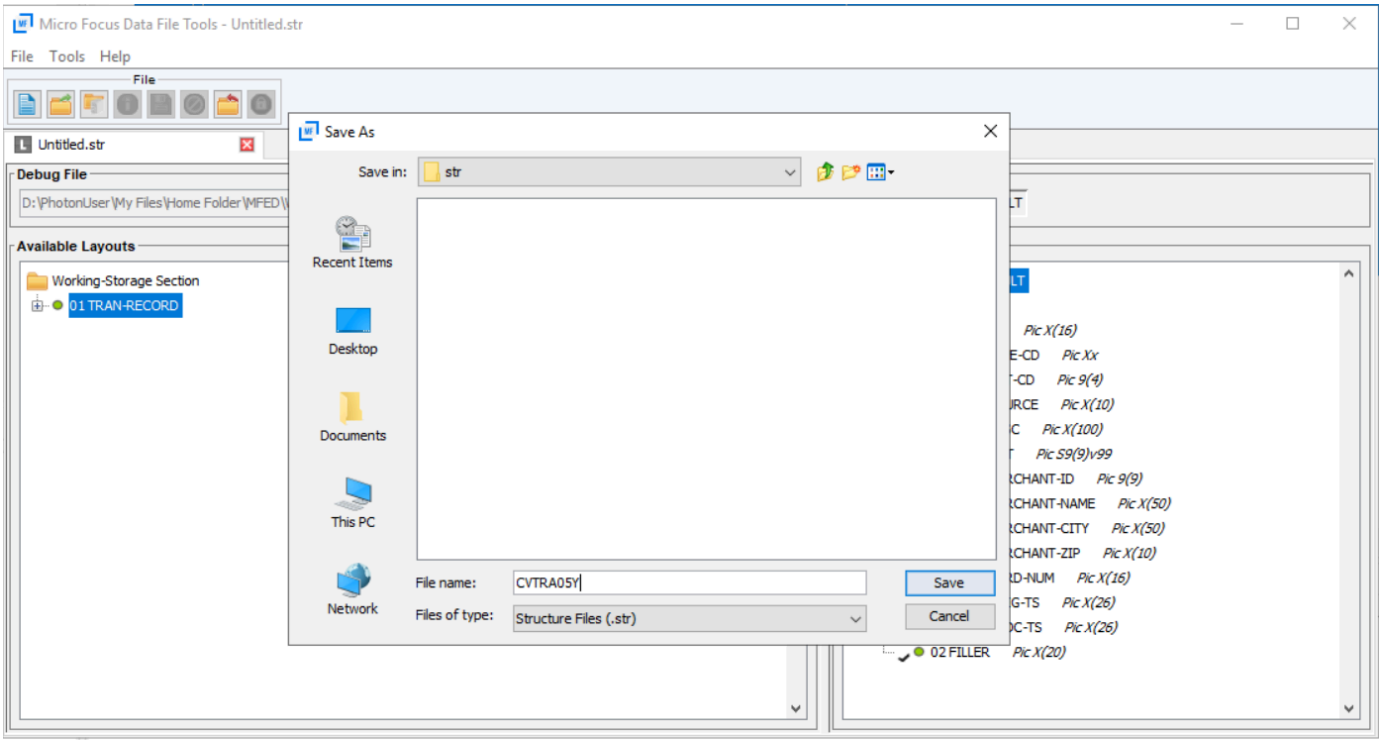


3. Fate clic con il pulsante destro del mouse sulla struttura del record e scegliete Crea layout predefinito (struttura singola) o Crea layout condizionale (struttura multipla) a seconda del layout.

   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. Sfoglia 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.

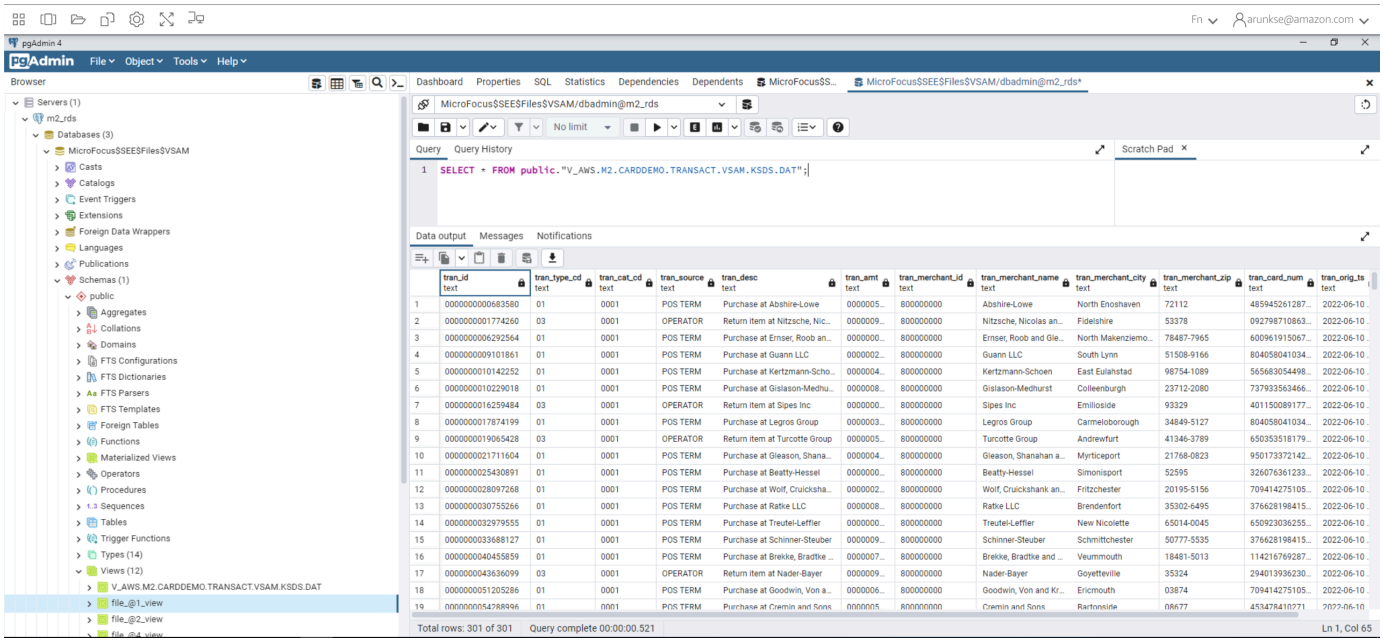
      VGN0017I Using structure definition 'TRAN-RECORD-DEFAULT'
      VGN0022I View 'V_AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT' installed in
 datastore 'sql://espacdatabase/VSAM'
      VGN0002I The operation completed successfully
```

Fase 5: Visualizzate i set di dati di Rocket Software (precedentemente Micro Focus) come tabelle e colonne

In questo passaggio, connettiti al database utilizzando in `pgAdmin` 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.TRANSACT.VSAM.KSDS.DAT";
```



Modifica i set di dati utilizzando gli strumenti per i file di dati di Rocket Software (precedentemente Micro Focus) in Enterprise Developer

È possibile visualizzare e modificare i set di dati in AWS Mainframe Modernization utilizzando il runtime Rocket Software per qualsiasi set di dati migrato.
I passaggi descritti in questo documento ti guideranno attraverso il processo di accesso ai set di dati utilizzando Data File Tools.
Ciò consente di visualizzare e modificare i set di dati migrati in base alle esigenze.

Argomenti

- Prerequisiti
- Avvia Rocket Software (precedentemente Micro Focus) Data File Tools
- Modifica i set di dati VSAM memorizzati nel database MFDBFH
- Modifica i set di dati non VSAM memorizzati nel database MFDBFH
- Modifica i set di dati VSAM e non VSAM memorizzati nel file system (EFS/FSx)

Prerequisiti

Prima di iniziare, è necessario disporre di un'applicazione distribuita con i set di dati importata nell'ambito del servizio AWS Mainframe Modernization utilizzando il motore Rocket Software.

Per continuare a modificare i set di dati, è necessario completare la Fase 1, la Fase 2, e (facoltativamente) il passaggio 3 dalla pagina the section called "Visualizza i set di dati come tabelle in Enterprise Developer" per configurare la connessione ODBC e il datastore Micro Focus (ovvero,). `MFDBFH`

> **Important**
>
> Questa guida presuppone che si stia utilizzando Amazon Aurora Postgres come datastore Micro Focus () `MFDBFH` per archiviare i dati dell'applicazione.

Avvia Rocket Software (precedentemente Micro Focus) Data File Tools

Dopo aver completato i prerequisiti, avviate Micro Focus Data File Tools impostando la variabile di `MFDBFH_CONFIG` ambiente per accedere ai set di dati memorizzati nel database (). `MFDBFH`

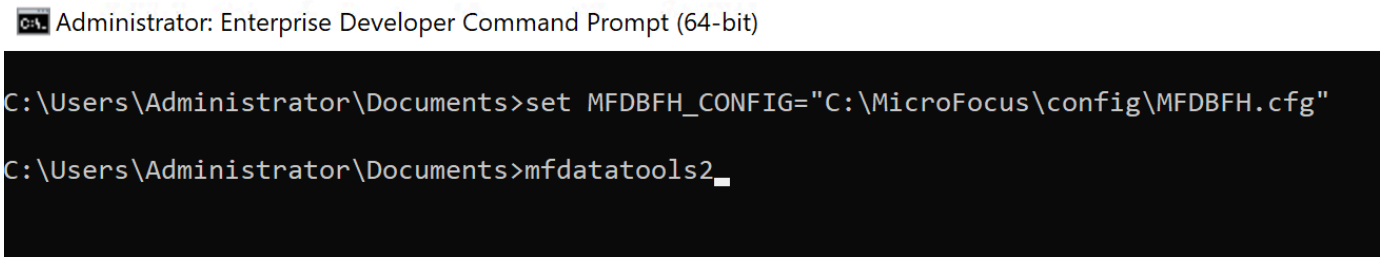
Per farlo:

1. Accedete al desktop Micro Focus Enterprise Developer e lanciate il prompt dei comandi di Enterprise Developer (64 bit) dal menu Start.
2. Imposta la variabile di `MFDBFH_CONFIG` ambiente con il percorso completo `MFDBCH.cfg` del file.

   ```
   set MFDBFH_CONFIG="C:\MicroFocus\config\MFDBFH.cfg"
   ```

3. Avviate Micro Focus Data File Tools dalla riga di comando di Enterprise Developer utilizzando il seguente comando.

   ```
   mfdatatools2
   ```



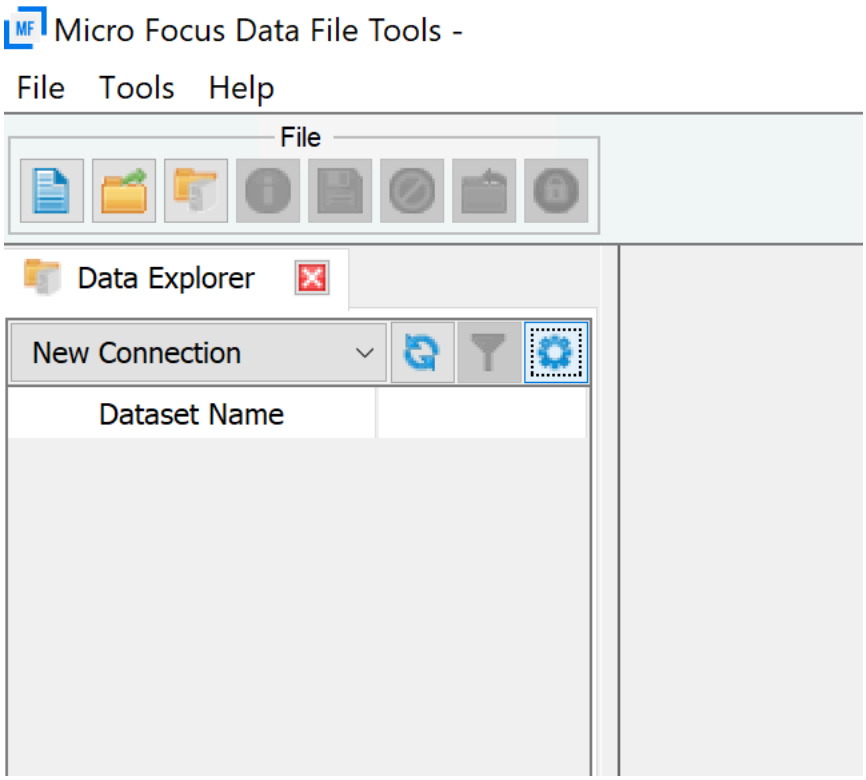
Questo apre Micro Focus Data File Tools in una finestra separata.

Modifica i set di dati VSAM memorizzati nel database MFDBFH

Una volta avviato Micro Focus Data File Tools, si apre un set di dati VSAM archiviato nel datastore Micro Focus.

Per farlo:

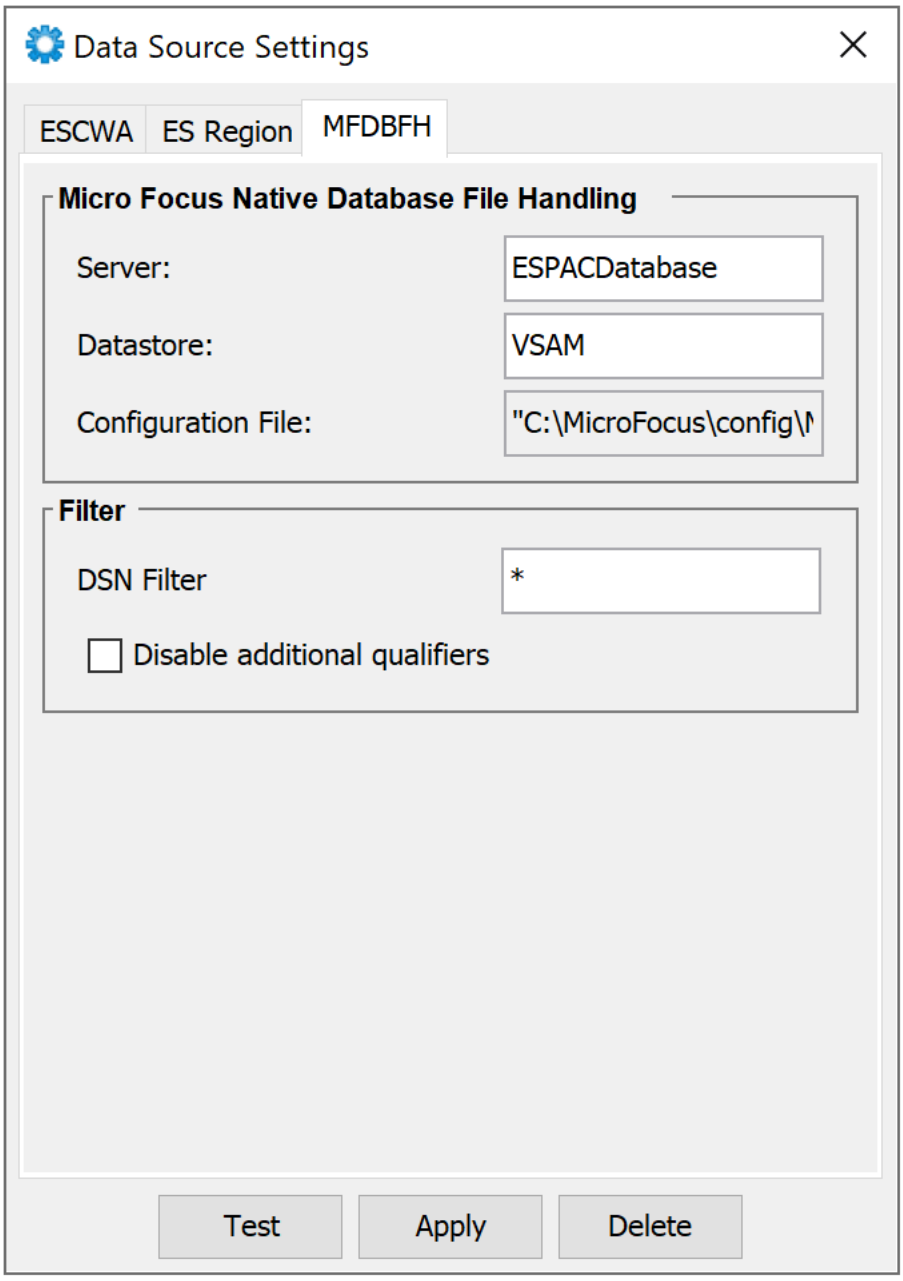
1. Dal menu File della finestra Micro Focus Data File Tools, scegliete Data Explorer.
2. Nella sezione Data Explorer, scegliete Impostazioni (icona a forma di ingranaggio) per configurare una nuova connessione. Si apre una finestra delle impostazioni dell'origine dati.



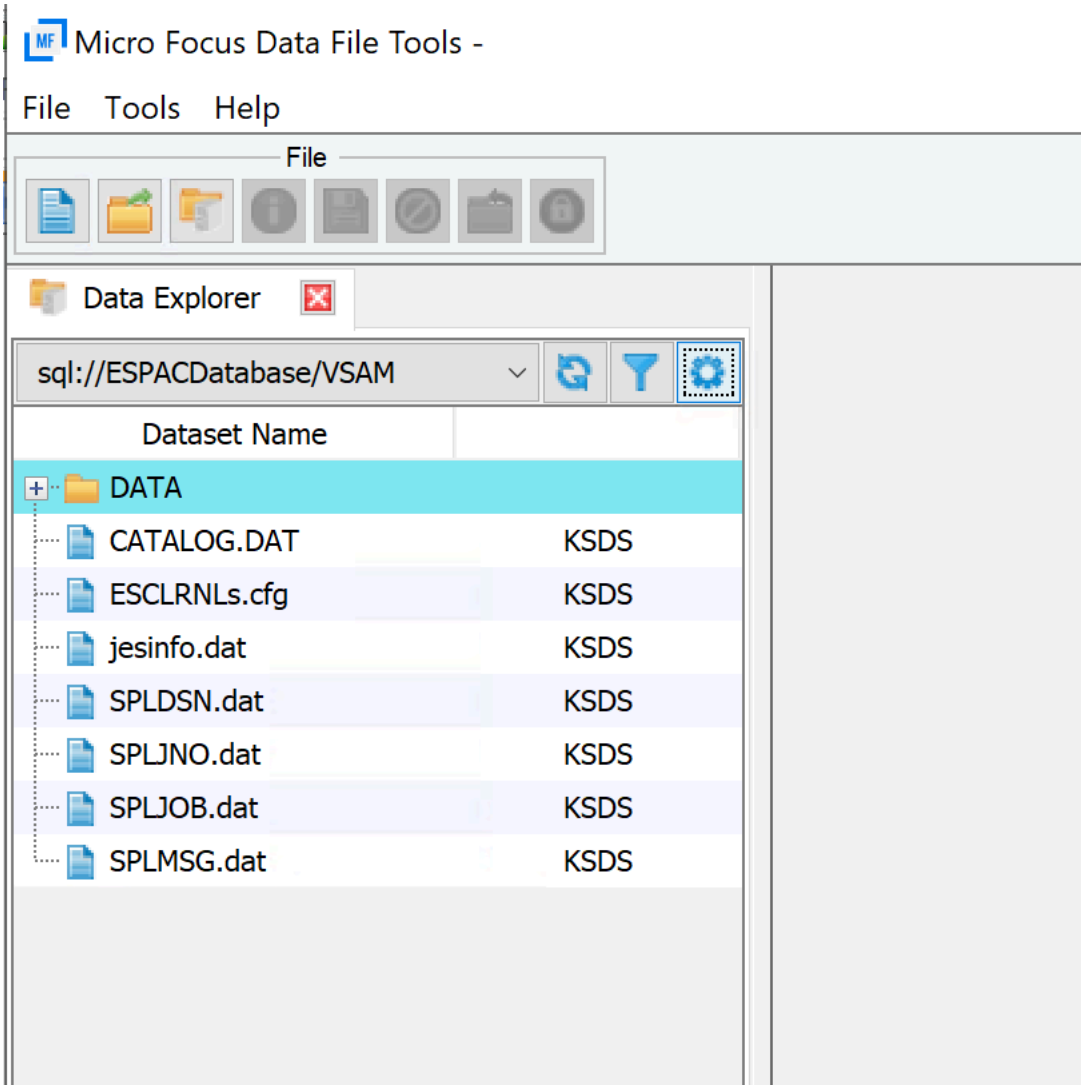
3. Nella finestra Data Source Settings, scegliete la scheda MFDBFH e immettete i seguenti valori:

- Server: `ESPACDatabase`
- Archivio dati: `VSAM`

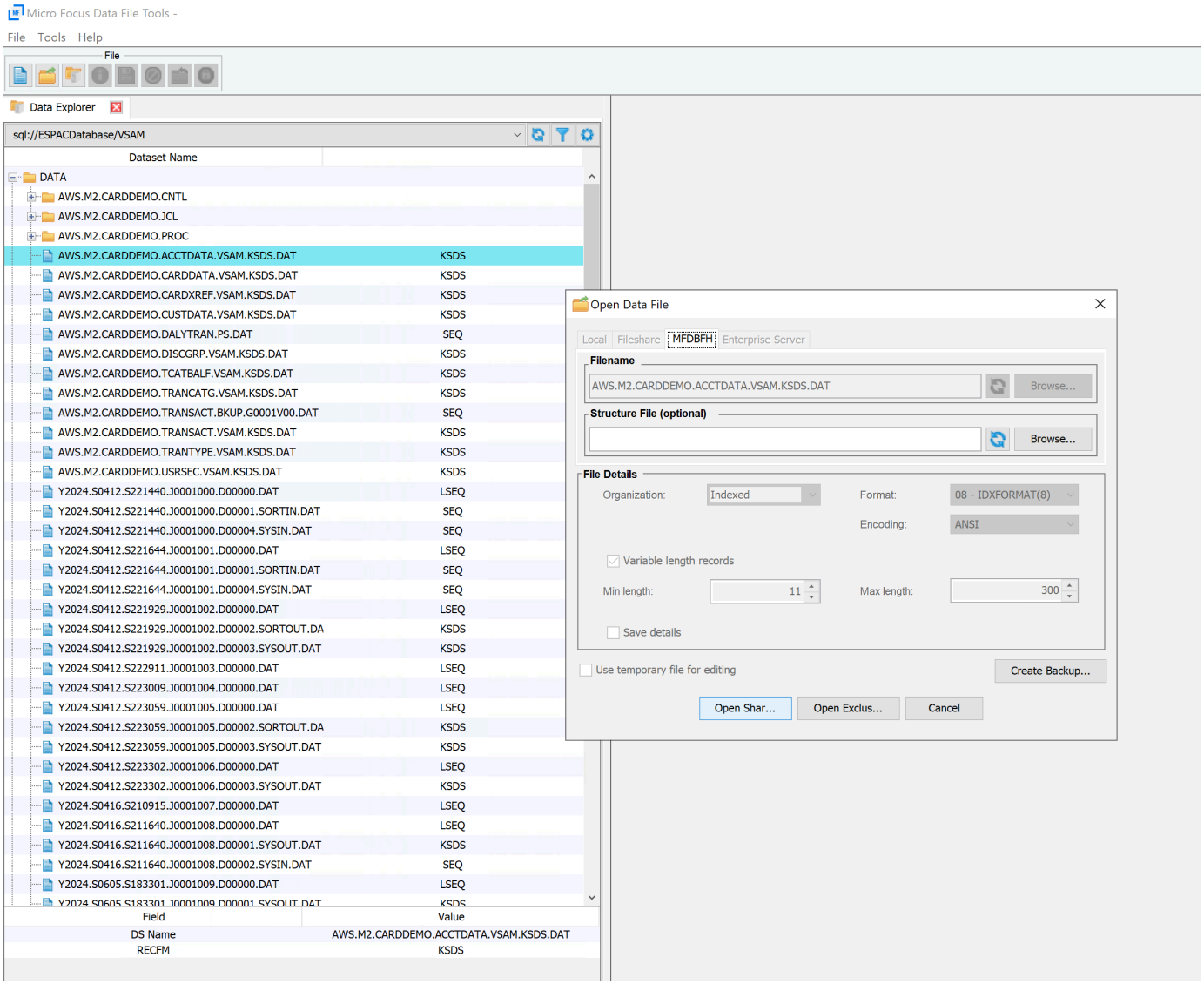
Scegli Applica per salvare la configurazione.



Data Explorer ora mostra tutti i set di dati memorizzati in. `MFDBFH`



4. Espandete il percorso relativo `DATA` e fate doppio clic sul set di dati VSAM che desiderate aprire.
5. Nella finestra Open Data File, scegli Apri condiviso o Apri esclusivo per aprire il set di dati.



È ora possibile visualizzare o modificare il set di dati aperto.

Modifica i set di dati non VSAM memorizzati nel database MFDBFH

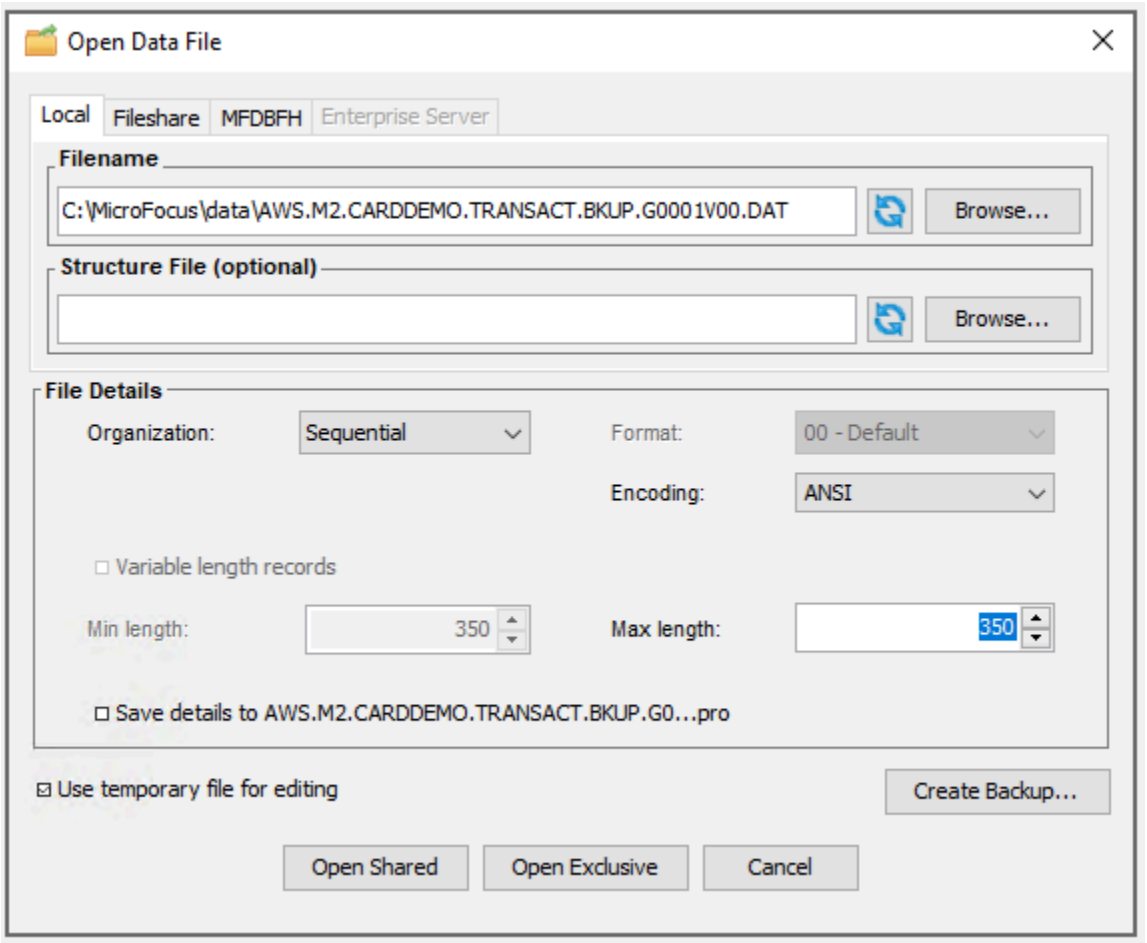
Se si desidera modificare set di dati non VSAM, si apre un set di dati non VSAM archiviato nel datastore Micro Focus.

Per farlo:

1. Dal prompt dei comandi di Enterprise Developer (64 bit), eseguite il `dbfhdeploy data extract` comando per scaricare il set di dati non VSAM nel file system locale.

> **Note**
>
> Prima di eseguire questo comando, assicuratevi di aver impostato la variabile di `MFDBFH_CONFIG` ambiente con il percorso completo del file. `MFDBFH.cfg`
>
> ```
> dbfhdeploy data extract sql://ESPACDatabase/VSAM/
> AWS.M2.CARDDEMO.TRANSACT.BKUP.G0001V00.DAT?folder=/DATA C:\MicroFocus\data
> \AWS.M2.CARDDEMO.TRANSACT.BKUP.G0001V00.DAT
> ```

2. Avviate Micro Focus Data File Tools dal menu Start.
3. Dal menu File di Micro Focus Data File Tools, scegliete Apri, quindi scegliete Data File.
4. Nella finestra Open Data File, sfogliate il set di dati scaricato nel file system locale. Modificate i dettagli del file come richiesto. Quindi scegli Open Shared o Open Exclusive per aprire il set di dati.



Ora puoi visualizzare o modificare il set di dati aperto.

I set di dati modificati o aggiornati possono essere reimportati nel datastore Micro Focus utilizzando i passaggi inclusi the section called "Importa set di dati per applicazioni" o utilizzando l'utilità da riga di comando dbfhdeploy.

Modifica i set di dati VSAM e non VSAM memorizzati nel file system (EFS/FSx)

È inoltre possibile aprire un set di dati archiviato in un file system.

Per farlo:

1. Installa il EFS/FSx file system sull'istanza Enterprise Developer EC2.
2. Utilizzate Micro Focus Data File Tools per sfogliare e aprire i set di dati dal file system.

Tutorial per Rocket Software (in precedenza Micro Focus)

I tutorial di questa sezione ti aiutano a iniziare a configurare varie attività nel motore di runtime di Rocket Software per il servizio di modernizzazione del AWS mainframe. Questi tutorial riguardano la configurazione di un'applicazione di esempio, l'utilizzo di modelli con Rocket Enterprise Developer e la configurazione di Enterprise Analyzer.

Argomenti

- Tutorial: configurazione della build Rocket Software (precedentemente Micro Focus) per l'applicazione di esempio BankDemo
- Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket Enterprise Developer
- Tutorial: utilizzo dei modelli con Rocket Enterprise Developer (precedentemente Micro Focus Enterprise Developer)
- Tutorial: configurare Enterprise Analyzer sulle applicazioni WorkSpaces
- Tutorial: configurare Rocket Enterprise Developer sulle applicazioni WorkSpaces

Tutorial: configurazione della build Rocket Software (precedentemente Micro Focus) per l'applicazione di esempio BankDemo

AWS La modernizzazione del mainframe offre la possibilità di configurare build e pipeline di integration/continuous distribuzione continua (CI/CD) per le applicazioni migrate. Queste build e pipeline utilizzano e forniscono queste 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 compila il codice sorgente, esegue test e produce pacchetti software pronti per la distribuzione. 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 AWS contattare specialisti del mainframe per conoscere i passaggi necessari per modernizzare un'applicazione mainframe.

Argomenti

- Prerequisiti
- Passaggio 1: condividi le risorse di compilazione con l' AWS account
- Fase 2: creazione di bucket Amazon S3
- Passaggio 3: Creare il file delle specifiche di compilazione
- Passaggio 4: carica i file sorgente
- Fase 5: Creare policy IAM

- Fase 6: Creare un ruolo IAM
- Fase 7: collegare le policy IAM al ruolo IAM
- Fase 8: Creare il progetto CodeBuild
- Fase 9: Avvia la compilazione
- Passaggio 10: scarica gli artefatti di output
- Eseguire la 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, i copybook e le definizioni. 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.

Passaggio 1: condividi le risorse di compilazione con l' AWS account

In questo passaggio, ti assicuri di condividere le risorse di compilazione con il tuo AWS account, specialmente nella regione in cui vengono utilizzate le risorse.

1. Apri la console di modernizzazione AWS del mainframe all'indirizzo. 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.

> **Important**
>
> Devi eseguire questo passaggio una volta in ogni AWS regione in cui intendi eseguire le build.

Fase 2: creazione di 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 si `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.

Passaggio 3: Creare il file delle specifiche di compilazione

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 CodeBuild utente.

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 4: 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 5: 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 Rocket Software. 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.

   JSON

```
{
    "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-*-<region>-prod"
        }
    ]
}
```

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.
2. Fornisci un nome per la politica, ad esempio,`BankdemoCodeBuildRolePolicy`.

Fase 6: Creare un ruolo IAM

In questo passaggio, crei un nuovo ruolo IAM che permetta di CodeBuild 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:

   JSON

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

9. Scegli Crea ruolo.

Fase 7: 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 8: 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,
   - Per Rocket Software v9: Invio. `673918848628.dkr.ecr.us-west-1.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1` Se utilizzi una AWS regione diversa con Rocket Software v9, puoi anche specificare dove si trova <m2-region>una AWS regione in cui è disponibile il servizio di modernizzazione del AWS mainframe (ad esempio `673918848628.dkr.ecr.<m2-region>.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1`,). `eu-west-3`
   - Per Rocket Software v8: Inserisci `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:8.0.9.R1`
   - Per Rocket Software v7: Invio `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:7.0.R10`
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 9: 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 10: scarica gli artefatti di output

In questo passaggio, scarichi gli artefatti di output da Amazon S3. Lo strumento di compilazione Rocket Software può creare diversi tipi di eseguibili. In questo tutorial, genera oggetti condivisi.

1. Accedi alla console Amazon S3.
2. Nella sezione Buckets role="bold">, scegli il nome del tuo bucket di output, ad esempio,. `codebuild-regionId-accountId-output-bucket`
3. Scegli Download role="bold">.

4. Decomprimere il file scaricato. Vai alla cartella di destinazione per vedere gli artefatti della build. Questi includono gli oggetti condivisi di `.so` Linux.

Eseguire la 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 WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket Enterprise Developer

AWS La modernizzazione del mainframe fornisce diversi strumenti tramite Amazon WorkSpaces Applications. WorkSpaces Applications è un servizio di streaming di applicazioni completamente gestito e sicuro che consente di trasmettere applicazioni desktop agli utenti senza riscrivere le applicazioni. WorkSpaces Applications 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 di WorkSpaces Applications to host di 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 del browser nelle WorkSpaces applicazioni, consulta i requisiti di sistema e il supporto delle funzionalità (browser Web) nella Amazon WorkSpaces Applications Administration Guide. In caso di problemi durante l'utilizzo WorkSpaces delle applicazioni, consulta la sezione Risoluzione dei problemi degli utenti AppStream 2.0 nella Amazon WorkSpaces Applications Administration Guide.

Questo documento è destinato ai membri del team operativo dei clienti. Descrive come configurare flotte e stack di Amazon WorkSpaces Applications per ospitare gli strumenti Rocket Enterprise

Analyzer e Rocket Enterprise Developer utilizzati con Mainframe Modernization. AWS Rocket Enterprise Analyzer viene solitamente utilizzato durante la fase di valutazione e Rocket Enterprise Developer viene solitamente utilizzato durante la fase di migrazione e modernizzazione dell'approccio di modernizzazione del mainframe. AWS Se si prevede di utilizzare sia Enterprise Analyzer che Enterprise Developer, è necessario creare flotte e stack separati per ogni strumento. Ogni strumento richiede una propria flotta e un proprio stack perché i termini di licenza sono diversi.

> **Important**
>
> I passaggi di questo tutorial si basano sul CloudFormation modello scaricabile cfn-m2- .yml. appstream-fleet-ea-ed

Argomenti

- Prerequisiti
- Passaggio 1: ottieni le immagini delle WorkSpaces applicazioni
- Passaggio 2: crea lo stack utilizzando il modello CloudFormation
- Passaggio 3: Creare un utente in WorkSpaces Applicazioni
- Fase 4: Accedere alle WorkSpaces applicazioni
- Passaggio 5: verifica i bucket in Amazon S3 (opzionale)
- Fasi successive
- Eseguire la pulizia delle risorse

Prerequisiti

- Scarica il modello: cfn-m2- .yml. appstream-fleet-ea-ed
- Ottieni l'ID del tuo VPC e del tuo gruppo di sicurezza predefiniti. Per ulteriori informazioni sul VPC predefinito, consulta Default VPCs nella Amazon VPC User Guide. Per ulteriori informazioni sul gruppo di sicurezza predefinito, consulta Gruppi di sicurezza predefiniti e personalizzati nella Guida per l'utente di Amazon EC2.
- Assicurati di disporre delle seguenti autorizzazioni:
  - crea pile, flotte e utenti nelle Applicazioni. WorkSpaces
  - crea pile CloudFormation utilizzando un modello.
  - crea bucket e carica file nei bucket in Amazon S3.

- scarica le credenziali (`access_key_idesecret_access_key`) da IAM.

Passaggio 1: ottieni le immagini delle WorkSpaces applicazioni

In questo passaggio, condividi WorkSpaces le immagini delle applicazioni per Enterprise Analyzer ed Enterprise Developer con il tuo AWS account.

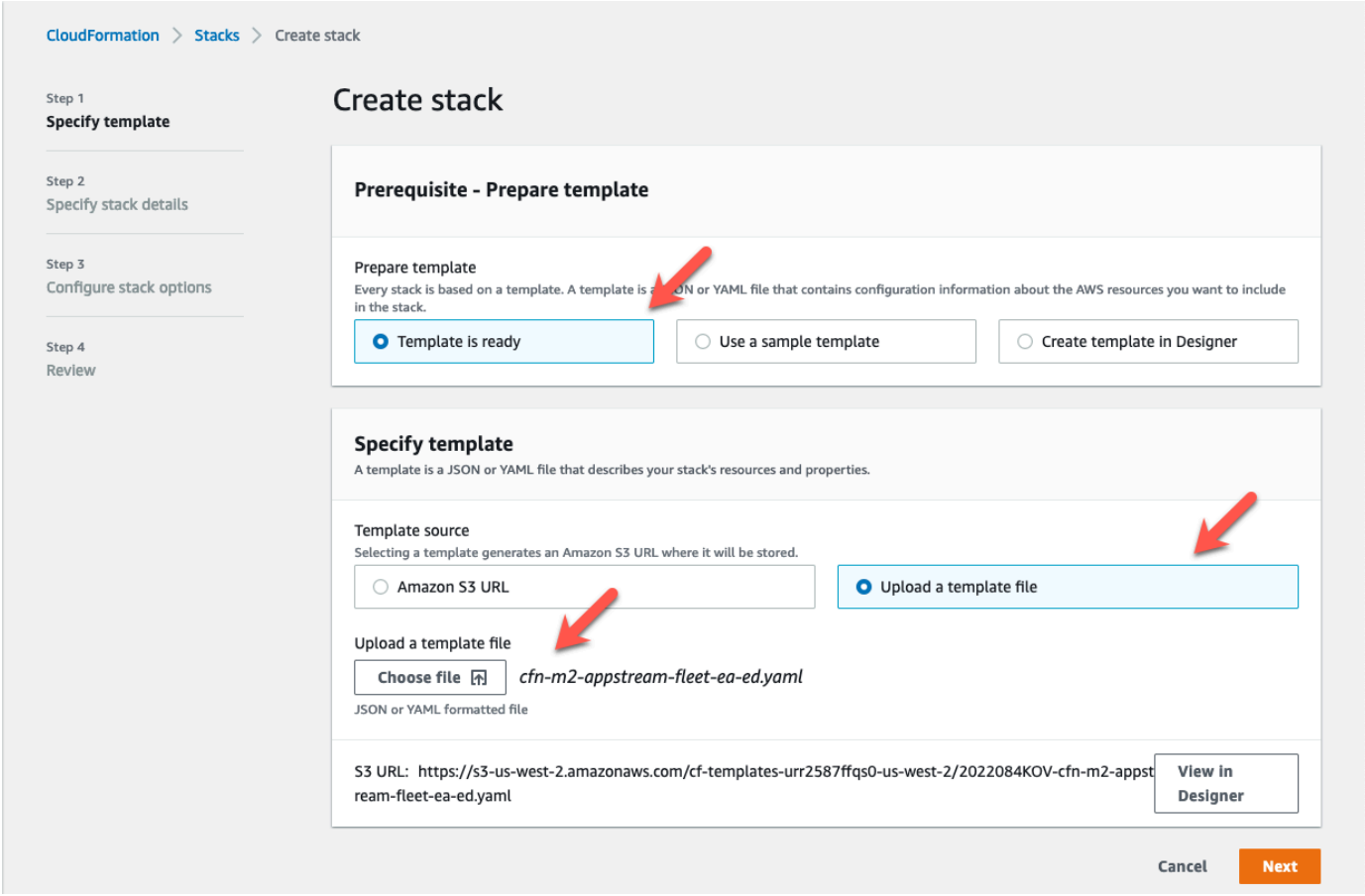
1. Apri la console di modernizzazione del AWS mainframe all'indirizzo. https://console.aws.amazon.com/m2/
2. Nella barra di navigazione a sinistra, scegli Strumenti.
3. In Analisi, sviluppo e creazione di risorse, scegli Condividi risorse con il mio AWS account.

Passaggio 2: crea lo stack utilizzando il modello CloudFormation

In questo passaggio, utilizzi il CloudFormation modello scaricato per creare uno stack e una flotta di WorkSpaces applicazioni per eseguire Rocket Enterprise Analyzer. Puoi ripetere questo passaggio in seguito per creare un altro stack e un altro parco WorkSpaces applicazioni per eseguire Rocket Enterprise Developer, poiché ogni strumento richiede il proprio parco applicazioni e il proprio stack. WorkSpaces Per ulteriori informazioni sugli CloudFormation stack, consulta Working with stacks nella Guida per l'utente.AWS CloudFormation

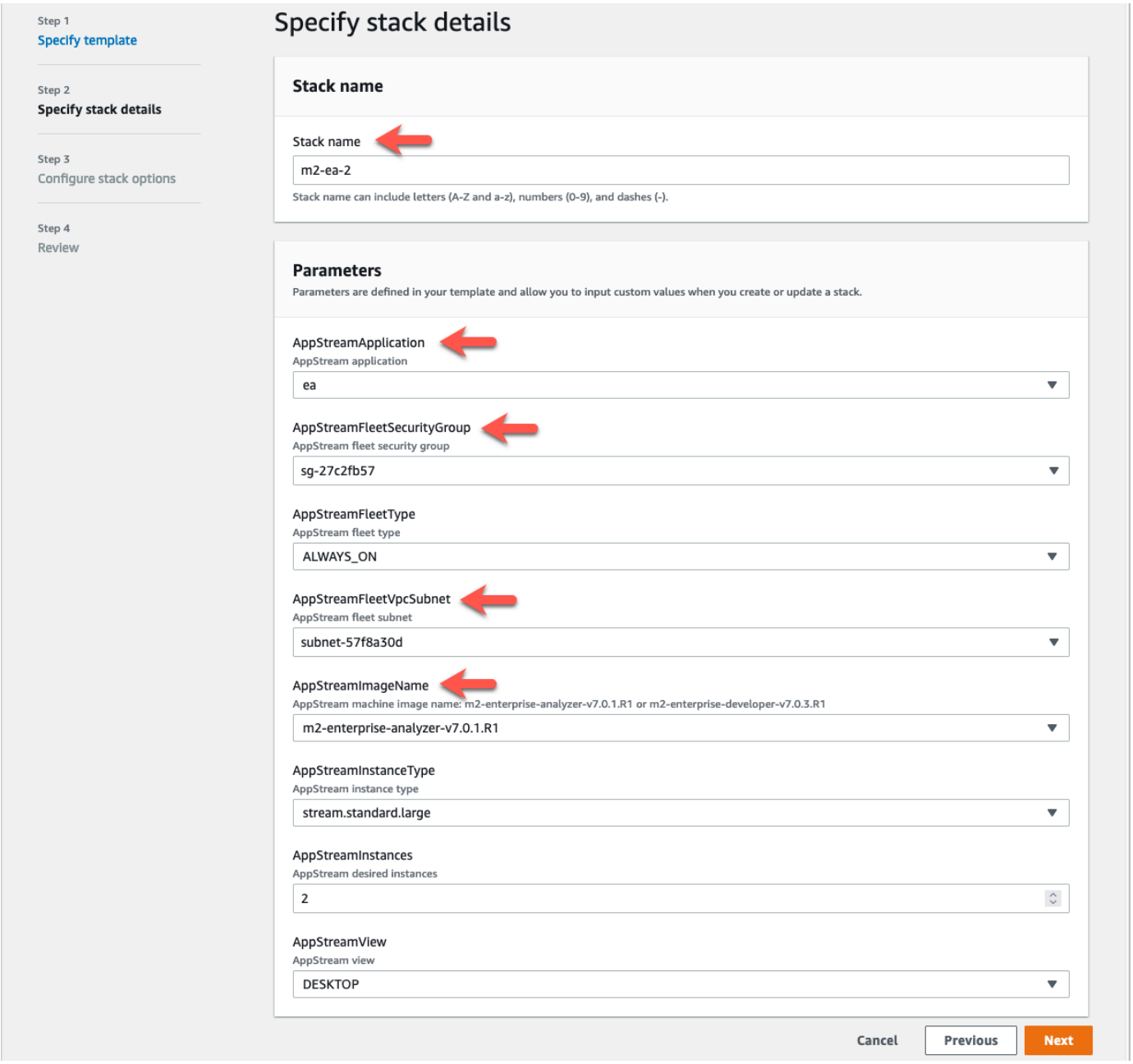
> **Note**
>
> AWS Mainframe Modernization aggiunge un costo aggiuntivo ai prezzi standard delle WorkSpaces applicazioni per l'uso di Enterprise Analyzer ed Enterprise Developer. Per ulteriori informazioni, consulta AWS Mainframe Modernization Pricing.

1. Scaricate il modello cfn-m2- appstream-fleet-ea-ed .yml, se necessario.
2. Apri la CloudFormation console e scegli Create Stack e con nuove risorse (standard).
3. In Prerequisito - Prepara modello, scegli Il modello è pronto.
4. In Specificare il modello, scegli Carica un file modello.
5. In Carica un file modello, scegli Scegli file e carica il modello cfn-m2- appstream-fleet-ea-ed .yml.
6. Scegli Next (Successivo).



7. In Specificare i dettagli dello stack, inserisci le seguenti informazioni:

   - Nel nome dello stack, inserisci un nome a tua scelta. Ad esempio, **`m2-ea`**.
   - In AppStreamApplication, scegli un tè.
   - In AppStreamFleetSecurityGroup, scegli il gruppo di sicurezza predefinito del tuo VPC predefinito.
   - In 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 Rocket Enterprise Analyzer.
   - Accetta le impostazioni predefinite per gli altri campi, quindi scegli Avanti.



8. Accetta tutte le impostazioni predefinite, quindi scegli nuovamente Avanti.
9. In Review, assicurati che tutti i parametri siano quelli che intendi.
10. Scorri verso il basso, 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 CloudFormation eventi man mano che si verificano.

Passaggio 3: Creare un utente in WorkSpaces Applicazioni

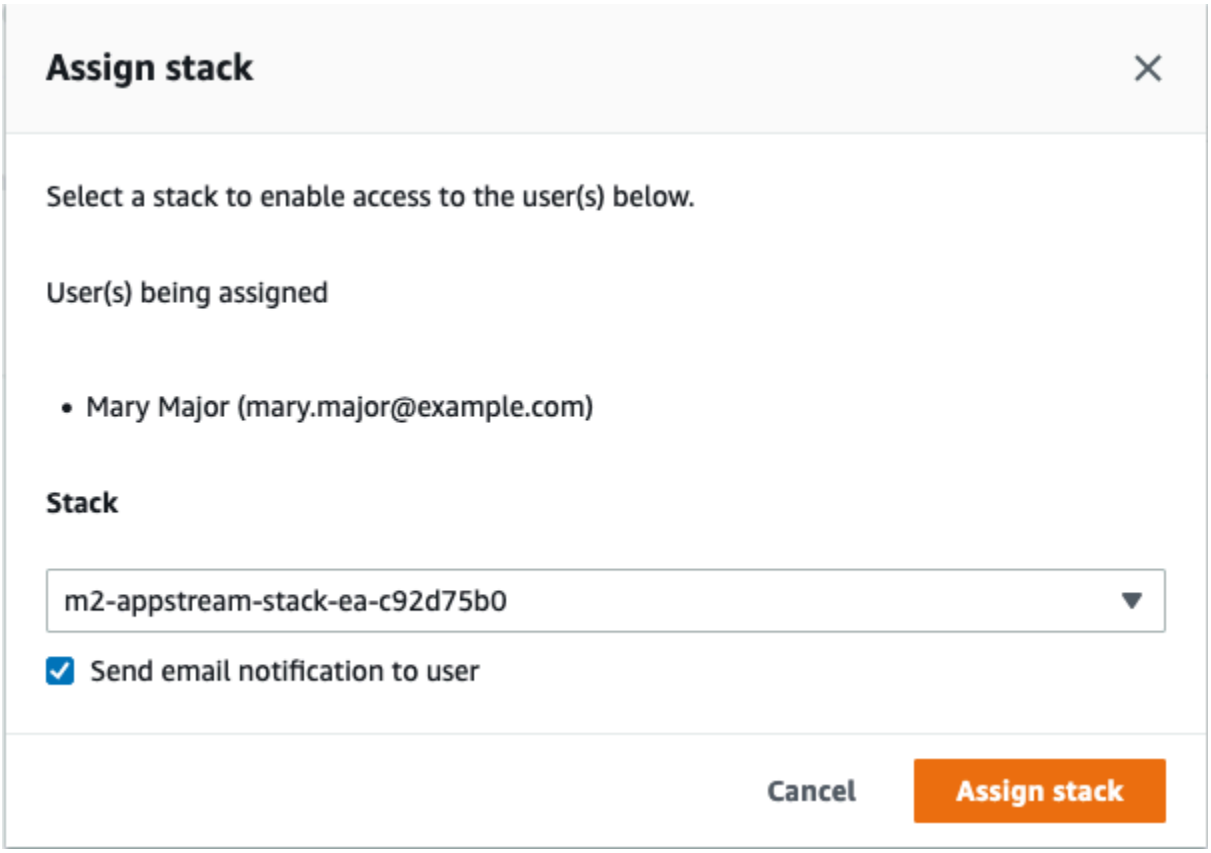
Mentre aspetti che la CloudFormation creazione dello stack sia completata, puoi creare uno o più utenti in WorkSpaces Applicazioni. Questi utenti sono coloro che utilizzeranno Enterprise Analyzer nelle WorkSpaces applicazioni. Dovrai specificare un indirizzo e-mail per ogni utente e assicurarti che ogni utente disponga di autorizzazioni sufficienti per creare bucket in Amazon S3, caricare file in un bucket e collegarsi a un bucket per mapparne il contenuto.

1. Apri la console Applicazioni. WorkSpaces
2. Nella barra di navigazione a sinistra, scegli User pool.
3. Selezionare Create user (Crea utente).
4. Fornisci un indirizzo e-mail a cui l'utente possa ricevere un invito via e-mail a utilizzare WorkSpaces le applicazioni, un nome e cognome e scegli Crea utente.
5. Ripeti 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 WorkSpaces delle applicazioni, consulta WorkSpaces Applications User Pools nella Amazon WorkSpaces Applications Administration Guide.

Al CloudFormation termine della creazione dello stack, puoi assegnare l'utente che hai creato allo stack, come segue:

1. Apri la console Applicazioni. WorkSpaces
2. Scegli il nome utente.
3. Scegli Azione, quindi Assegna stack.
4. In Assegna stack, scegli lo stack che inizia con. `m2-appstream-stack-ea`
5. Scegliere Assign stack (Assegna stack).



L'assegnazione di un utente a uno stack fa sì che WorkSpaces Applications invii un'e-mail all'utente all'indirizzo fornito. Questa e-mail contiene un collegamento alla pagina di accesso alle WorkSpaces Applicazioni.

Fase 4: Accedere alle WorkSpaces applicazioni

In questo passaggio, accedi alle WorkSpaces Applicazioni utilizzando il link contenuto nell'e-mail inviata da WorkSpaces Applications all'utente in cui hai creatoPassaggio 3: Creare un utente in WorkSpaces Applicazioni.

1. Accedi alle WorkSpaces Applicazioni utilizzando il link fornito nell'e-mail inviata dalle WorkSpaces Applicazioni.
2. Cambia la password, se richiesto. La schermata WorkSpaces Applicazioni che vedi è simile alla seguente:



3. Scegli Desktop.
4. Sulla barra delle applicazioni, scegli Cerca ed entra **`D:`** per accedere alla cartella Home.

> **Note**
>
> Se salti questo passaggio, potresti ricevere un `Device not ready` errore quando tenti di accedere alla cartella Home.

In qualsiasi momento, se riscontri problemi di accesso WorkSpaces alle Applicazioni, puoi riavviare il parco WorkSpaces Applicazioni e riprovare ad accedere, utilizzando la procedura seguente.

1. Apri la console WorkSpaces Applicazioni.
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. Attendi 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 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 tra le sessioni di lavoro. Questi bucket sono i seguenti:

- Il nome inizia con`appstream2-`. Questo bucket mappa i dati nella cartella Home in WorkSpaces Applicazioni (`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 account. AWS Il nome della Home Folder è un SHA256 hash dell'indirizzo e-mail dell'utente ed è memorizzato in un percorso basato su tale hash.

- Il nome inizia con. `appstream-app-settings-` Questo bucket contiene informazioni sulla sessione utente per WorkSpaces le applicazioni 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 WorkSpaces Applications Administration Guide.

Per verificare che i bucket siano stati creati, segui questi passaggi:

1. Apri la console Amazon S3.
2. Nella barra di navigazione a sinistra, scegli Bucket.
3. In Trova i bucket per nome, inserisci **`appstream`** per filtrare l'elenco.

Se vedi i bucket, non sono necessarie ulteriori azioni. Tieni solo presente che i bucket esistono. Se non vedi i bucket, significa che l'esecuzione del CloudFormation modello non è terminata o si è verificato un errore. Vai alla CloudFormation console e rivedi i messaggi di creazione dello stack.

Fasi successive

Ora che l'infrastruttura WorkSpaces delle applicazioni è configurata, è possibile configurarla e iniziare a utilizzare Enterprise Analyzer. Per ulteriori informazioni, consulta Tutorial: configurare Enterprise Analyzer sulle applicazioni WorkSpaces. È inoltre possibile configurare Enterprise Developer. Per ulteriori informazioni, consulta Tutorial: configurare Rocket Enterprise Developer sulle applicazioni WorkSpaces.

Eseguire la pulizia delle risorse

La procedura per ripulire lo stack e le flotte creati è descritta in Creare un parco macchine e uno stack di WorkSpaces applicazioni.

Quando gli oggetti WorkSpaces Applications sono stati eliminati, l'amministratore dell'account può anche, se necessario, pulire i bucket Amazon S3 per le impostazioni delle applicazioni 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 di WorkSpaces applicazioni.

Infine, attualmente WorkSpaces Applications non consente di eliminare gli utenti utilizzando la console. È invece necessario utilizzare l'API del servizio con la CLI. Per ulteriori informazioni, consulta User Pool Administration nella Amazon WorkSpaces Applications Administration Guide.

Tutorial: utilizzo dei modelli con Rocket Enterprise Developer (precedentemente Micro Focus Enterprise Developer)

Questo tutorial descrive come utilizzare modelli e progetti predefiniti con Rocket Enterprise Developer. Copre tre casi d'uso. Tutti i casi d'uso utilizzano il codice di esempio fornito nell' BankDemo esempio. Per scaricare l'esempio, scegli `bankdemo.zip`.

> **Important**
>
> Se utilizzate la versione di Enterprise Developer per Windows, i file binari generati dal compilatore possono essere eseguiti solo sull'Enterprise Server fornito con Enterprise Developer. Non è possibile eseguirli nel runtime AWS Mainframe Modernization, che è basato su Linux.

Argomenti

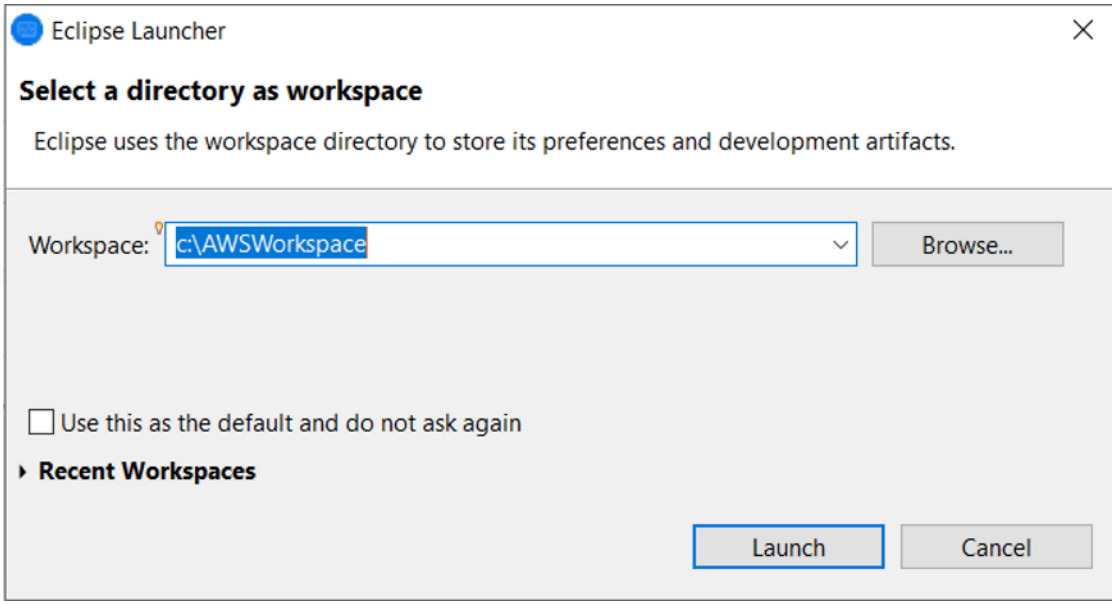
- Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine

- Caso d'uso 2: utilizzo del modello di progetto COBOL senza componenti di origine
- Caso d'uso 3: utilizzo del progetto COBOL predefinito che si collega alle cartelle di origine
- Utilizzo del modello JSON Region Definition

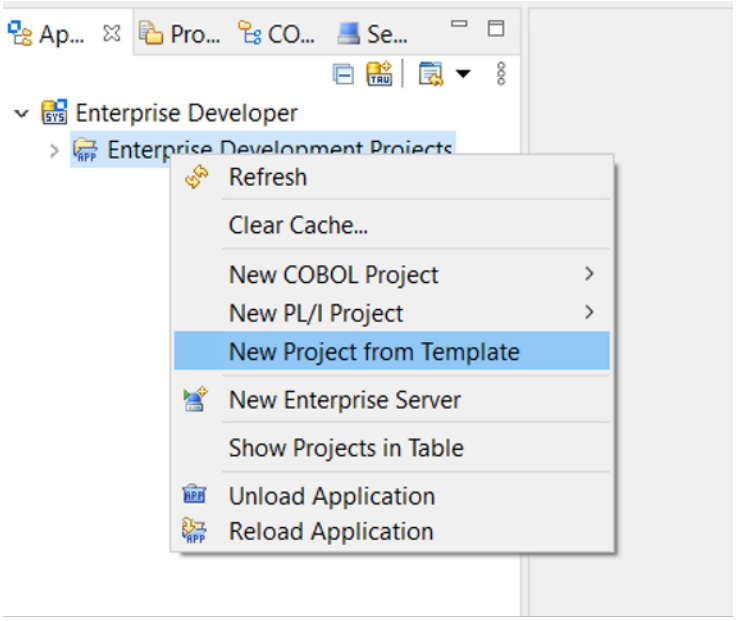
Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine

Questo caso d'uso richiede la copia dei componenti di origine nella struttura di directory Template come parte dei passaggi preliminari alla configurazione dimostrativa. Nel frattempo `bankdemo.zip`questo è stato modificato rispetto alla versione originale `AWSTemplates.zip` per evitare di avere due copie del codice sorgente.

1. Avvia Enterprise Developer e specifica l'area di lavoro scelta.

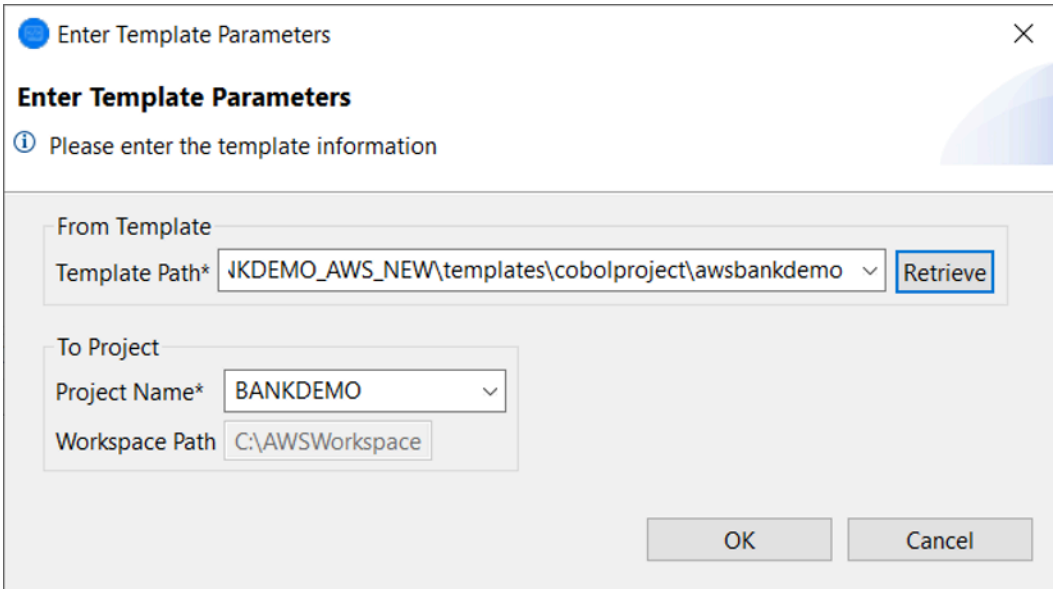


2. Nella vista Application Explorer, dalla voce di visualizzazione ad albero di Enterprise Development Project, scegliete Nuovo progetto da modello dal menu contestuale.

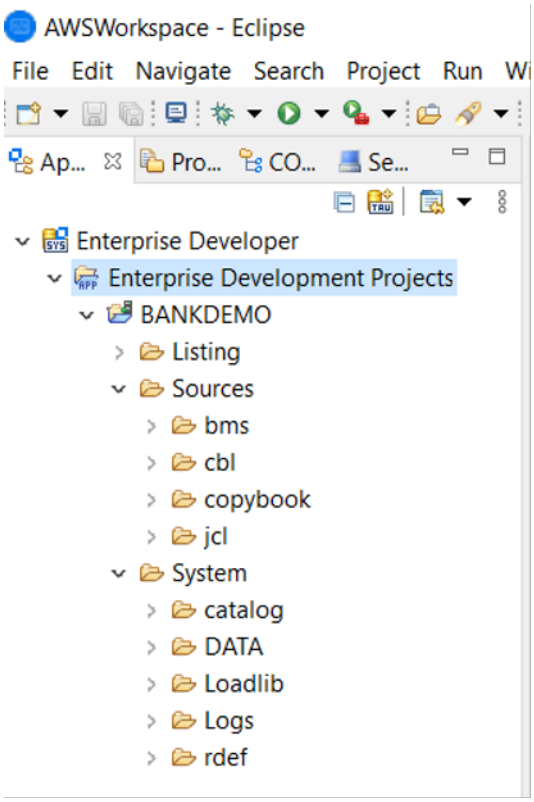


3. Immettete i parametri del modello come mostrato.

> **Note**
>
> Il percorso del modello farà riferimento a dove è stato estratto lo ZIP.



4. Scegliendo OK verrà creato un progetto Eclipse di sviluppo locale basato sul modello fornito, con una struttura completa di origine e ambiente di esecuzione.



La `System` struttura contiene un file completo di definizione delle risorse con le voci richieste per BANKDEMO, il catalogo richiesto con le 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 quindi vengono automaticamente integrati in Enterprise Developer.

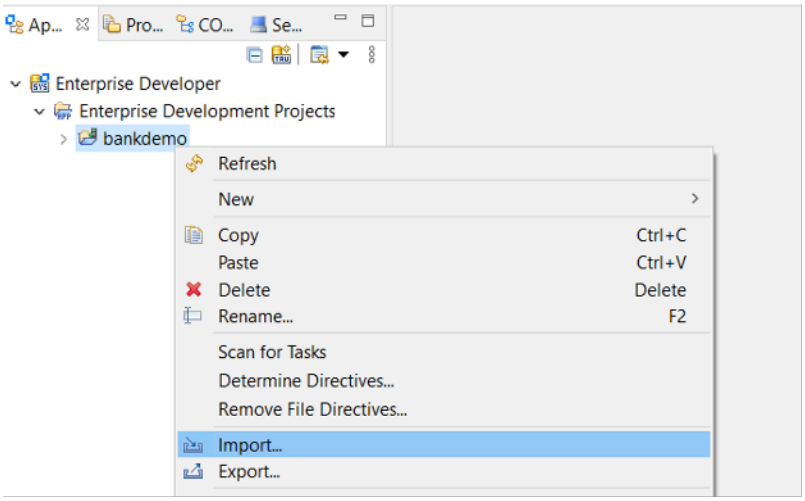
Caso d'uso 2: utilizzo del modello di progetto COBOL senza componenti di origine

I passaggi da 1 a 3 sono identici a. Caso d'uso 1: utilizzo del modello di progetto COBOL contenente i componenti di origine

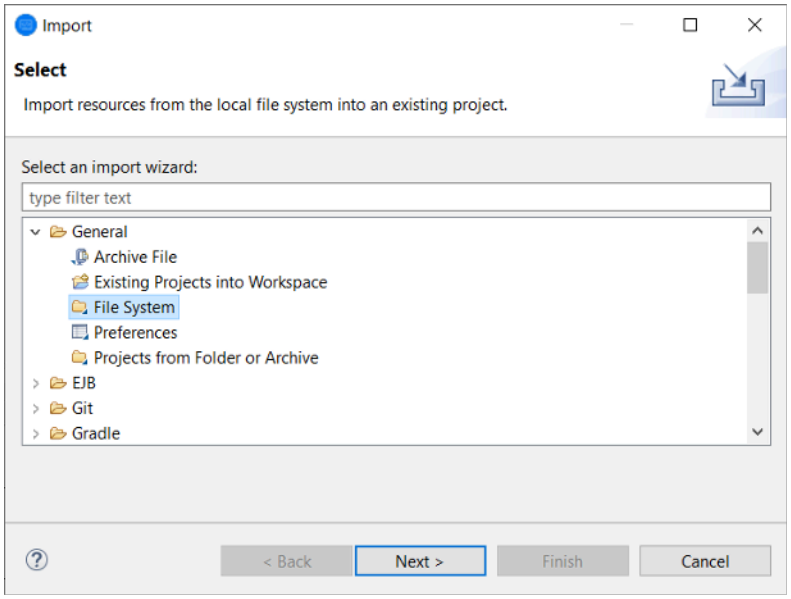
La `System` struttura in questo caso d'uso contiene anche un file completo di definizione delle risorse con le voci richieste per BankDemo, il catalogo richiesto con le voci aggiunte e i corrispondenti file di dati ASCII.

Tuttavia, la struttura di origine del modello non contiene alcun componente. È necessario importarli nel progetto da qualsiasi repository di origine che si stia utilizzando.

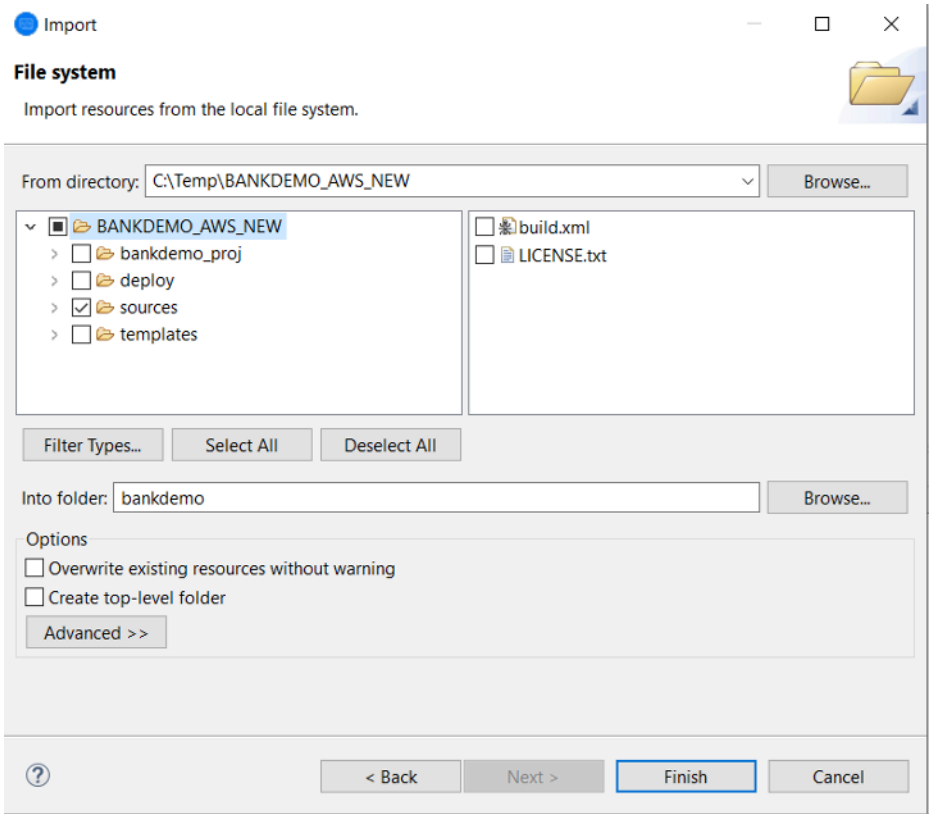
1. Scegli il nome del progetto. Dal menu contestuale correlato, scegliete Importa.



2. Nella finestra di dialogo risultante, nella sezione Generale, scegliete File System, quindi scegliete Avanti.



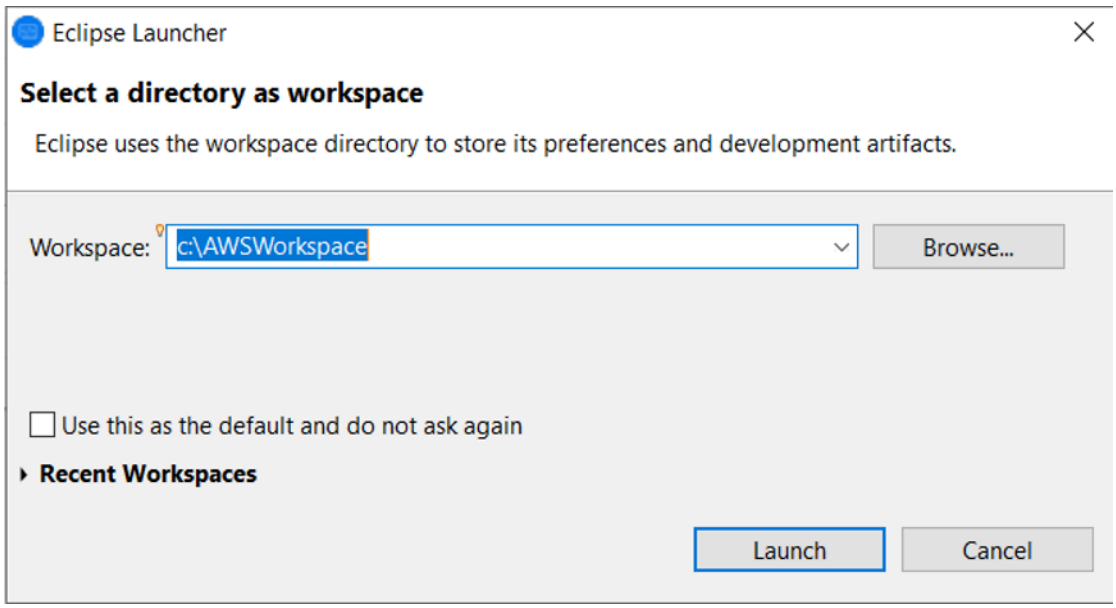
3. Compila il campo Dalla directory sfogliando il file system in modo che punti alla cartella del repository. Scegli tutte le cartelle che desideri importare, ad esempio. `sources` Il `Into folder` campo verrà precompilato. Scegli Fine.



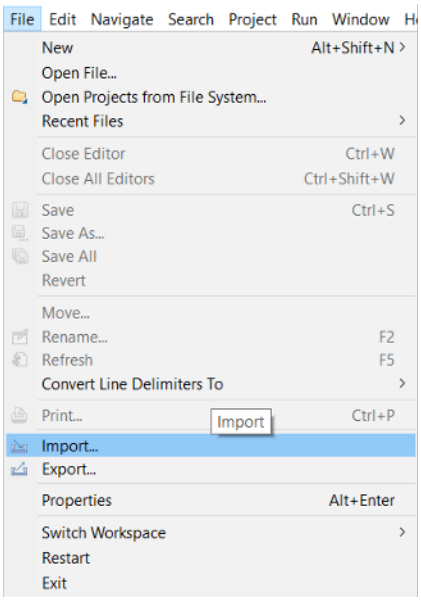
Dopo che la struttura del modello di origine contiene tutti gli elementi di origine, questi vengono creati automaticamente in Enterprise Developer.

Caso d'uso 3: utilizzo del progetto COBOL predefinito che si collega alle cartelle di origine

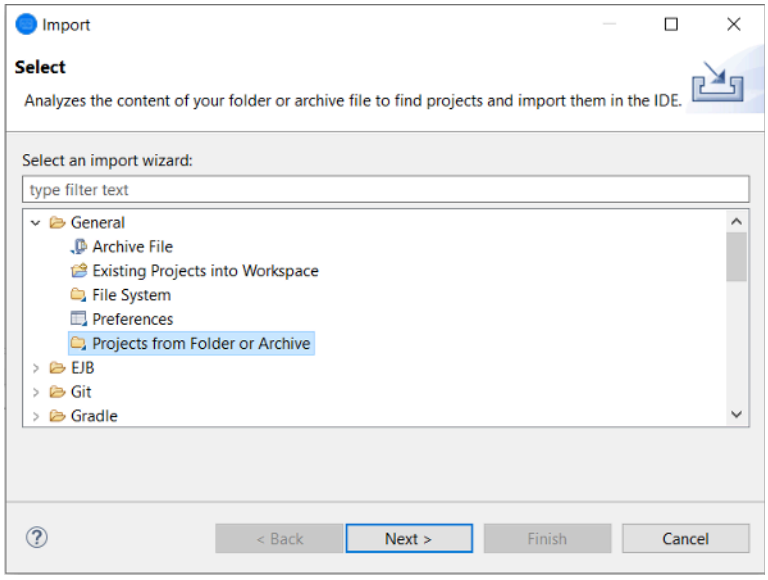
1. Avvia Enterprise Developer e specifica l'area di lavoro scelta.



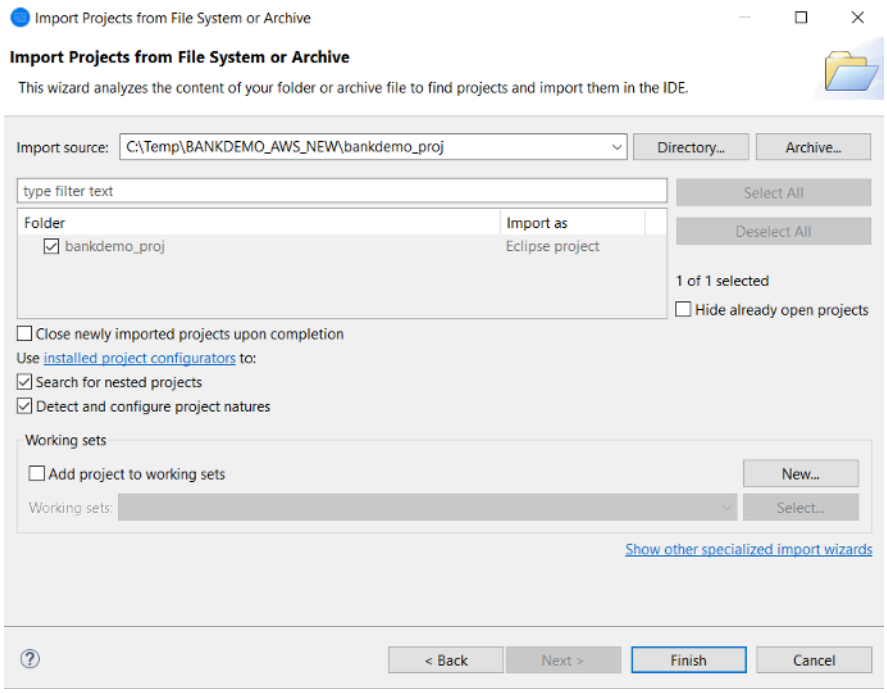
2. Dal menu File scegliere Import (Importa).



3. Nella finestra di dialogo risultante, in Generale, scegli Progetti da cartella o Archivio e scegli Avanti.



4. Compilate la sorgente di importazione, scegliete Directory e sfogliate il file system per selezionare la cartella di progetto predefinita. Il progetto contenuto all'interno contiene collegamenti alle cartelle di origine nello stesso repository.

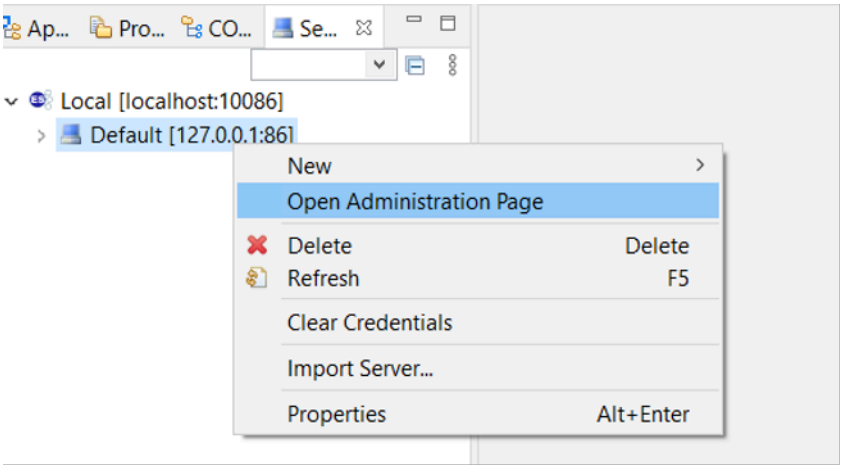


Scegli Fine.

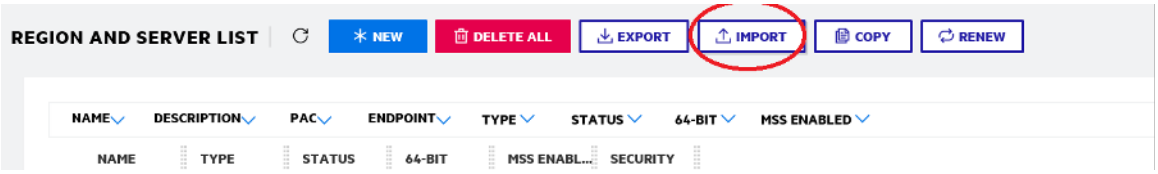
Poiché il progetto è popolato dai collegamenti alla cartella di origine, il codice viene creato automaticamente.

Utilizzo del modello JSON Region Definition

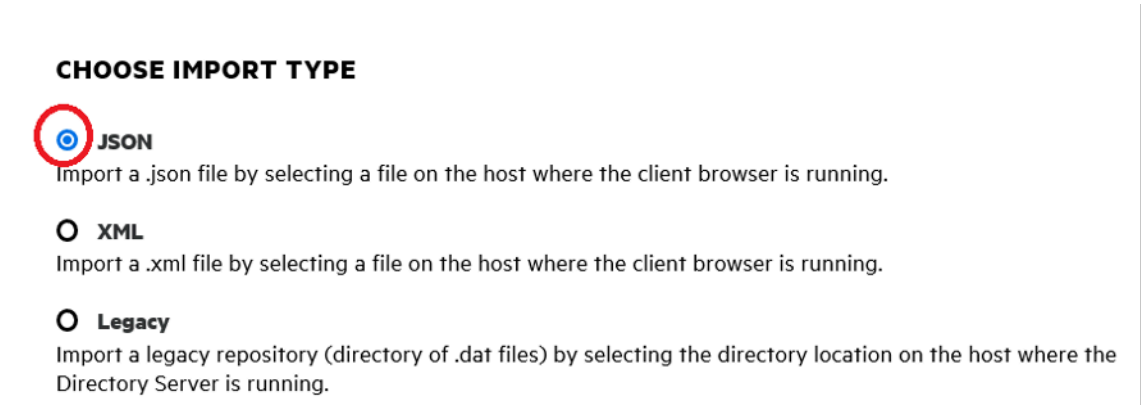
1. Passa alla visualizzazione Server Explorer. Dal menu contestuale correlato, scegliete Apri pagina di amministrazione, che avvia il browser predefinito.



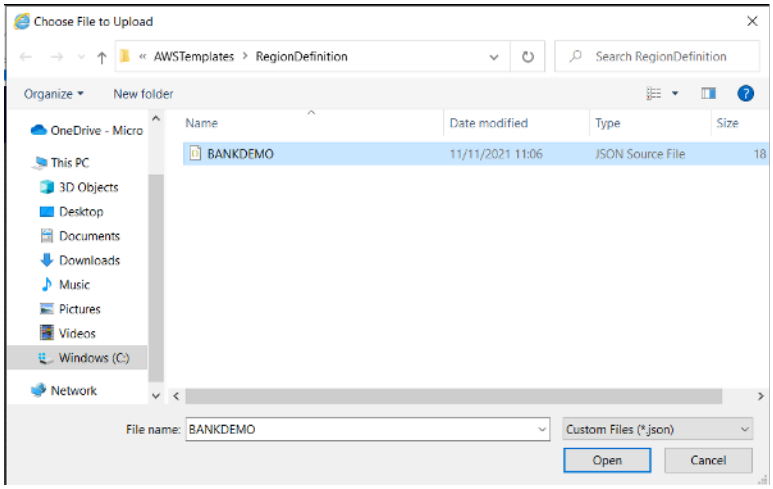
2. Dalla schermata risultante di Enterprise Server Common Web Administration (ESCWA), scegliete Importa.



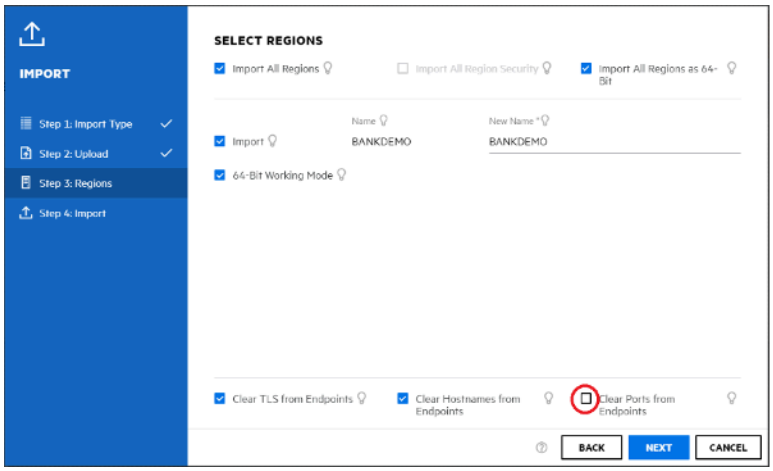
3. Scegli il tipo di importazione JSON e scegli Avanti.



4. Carica il `BANKDEMO.JSON` file fornito.

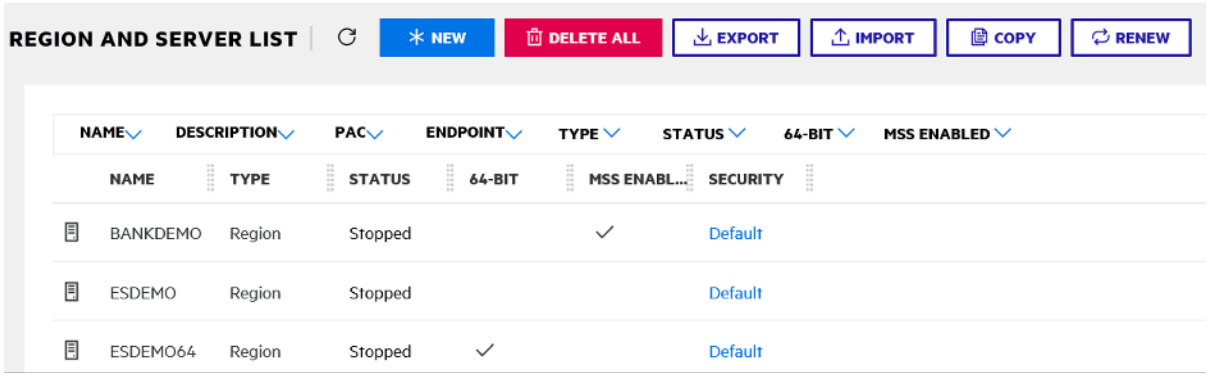


Una volta selezionato, scegli Avanti.

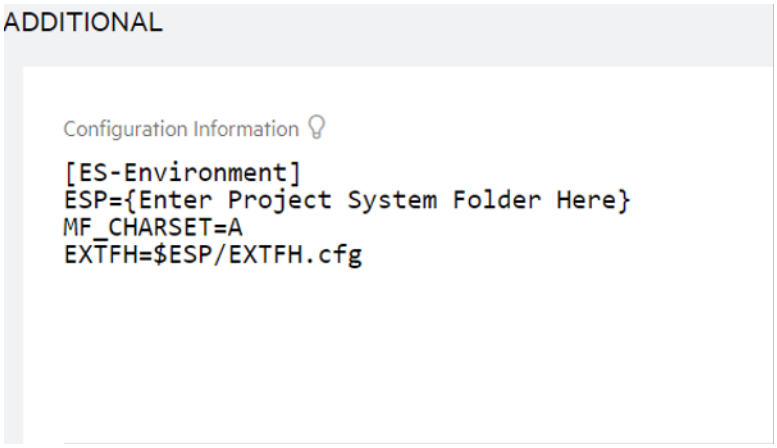


Nel pannello Seleziona regioni, assicuratevi che l'opzione Cancella porte dagli endpoints non sia selezionata, quindi continuate a scegliere Avanti tra i pannelli finché non viene visualizzato il pannello Esegui importazione. Quindi scegliete Importa dal riquadro di navigazione a sinistra.

Infine, fai clic su Fine. La regione BANKDEMO verrà quindi aggiunta all'elenco dei server.

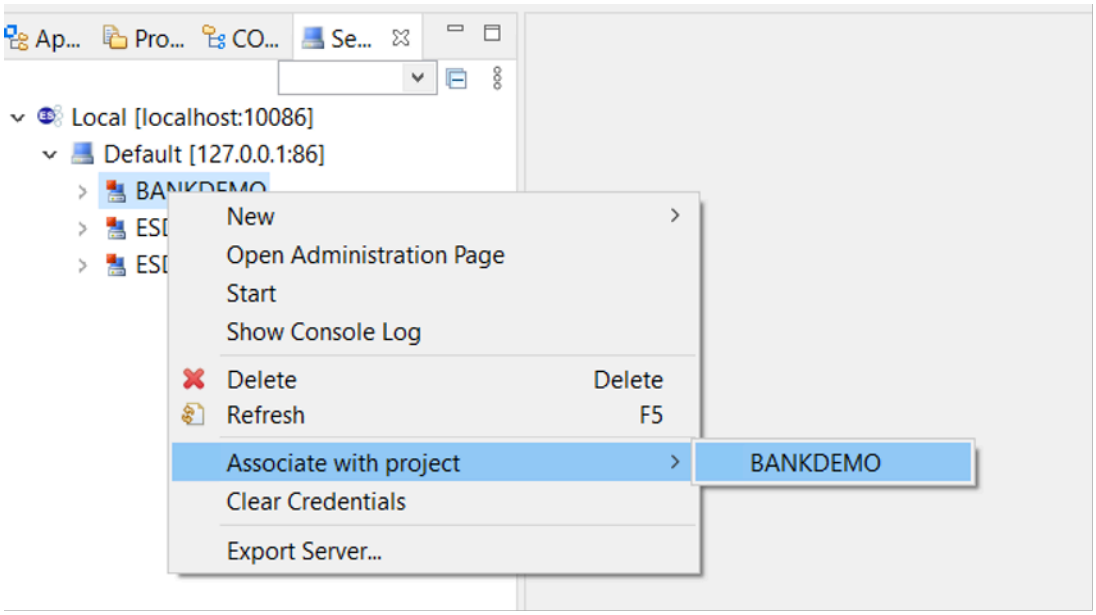


5. Vai alle Proprietà generali della regione BANKDEMO.
6. Scorri fino alla sezione Configurazione.
7. La variabile di ambiente ESP deve essere impostata sulla `System` cartella relativa al progetto Eclipse creato nei passaggi precedenti. Questo dovrebbe essere. `workspacefolder/projectname/System`



8. Fare clic su Apply (Applica).

   La regione è ora completamente configurata per funzionare insieme al progetto Eclipse COBOL.
9. Infine, tornando in Enterprise Developer, associate 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 in base alla regione.

> **Important**
>
> Se si utilizza la versione di Enterprise Developer per Windows, i file binari generati dal compilatore possono essere eseguiti solo sull'Enterprise Server fornito con Enterprise Developer. Non è possibile eseguirli nel runtime AWS Mainframe Modernization, che è basato su Linux.

Tutorial: configurare Enterprise Analyzer sulle applicazioni WorkSpaces

Questo tutorial descrive come configurare Rocket Enterprise Analyzer (precedentemente 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 delle WorkSpaces applicazioni che AWS Mainframe Modernization condivide con il cliente. L'installazione si basa inoltre sulla creazione di flotte e stack di WorkSpaces applicazioni, come descritto in. Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket Enterprise Developer

> **Important**
>
> I passaggi di questo tutorial presuppongono la configurazione WorkSpaces delle applicazioni con il CloudFormation modello scaricabile cfn-m2- .yml. appstream-fleet-ea-ed Per ulteriori informazioni, consulta Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket 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 di Rocket Software (precedentemente Micro Focus).

Contenuto dell'immagine

Oltre all'applicazione Enterprise Analyzer stessa, l'immagine contiene i seguenti strumenti e librerie.

Strumenti di terze parti

- Python
- Reclone
- 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
- Fase 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
- Eseguire la 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 DB2 oggetti 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 AWSPer ulteriori informazioni, vedere Accesso programmatico in.Riferimenti generali di Amazon Web Services

Fase 1: configurazione

1. Avvia una sessione con WorkSpaces Applications con l'URL che hai ricevuto nel messaggio e-mail di benvenuto inviato da WorkSpaces Applications.
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 WorkSpaces Applicazioni, scegli Desktop per accedere al desktop Windows su cui è in streaming la flotta.

Fase 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 `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 `amzn-s3-demo-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 alle WorkSpaces applicazioni 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:amzn-s3-demo-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:

   Description

   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. Accedere al repository nel riquadro di sinistra, selezionare il nome del repository e scegliere Aggiungi file/cartelle allo spazio di lavoro. Seleziona 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. Avvia una sessione con WorkSpaces Applications con l'URL che hai ricevuto nel messaggio e-mail di benvenuto inviato da WorkSpaces Applications.
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.

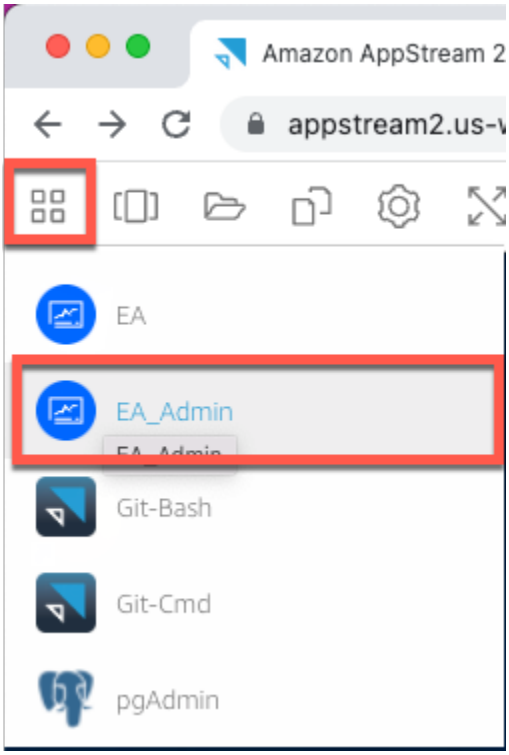
Risoluzione dei problemi di connessione all'area di lavoro

Quando si tenta di riconnettersi all'area di lavoro di Enterprise Analyzer, è possibile che venga visualizzato un errore simile al seguente:

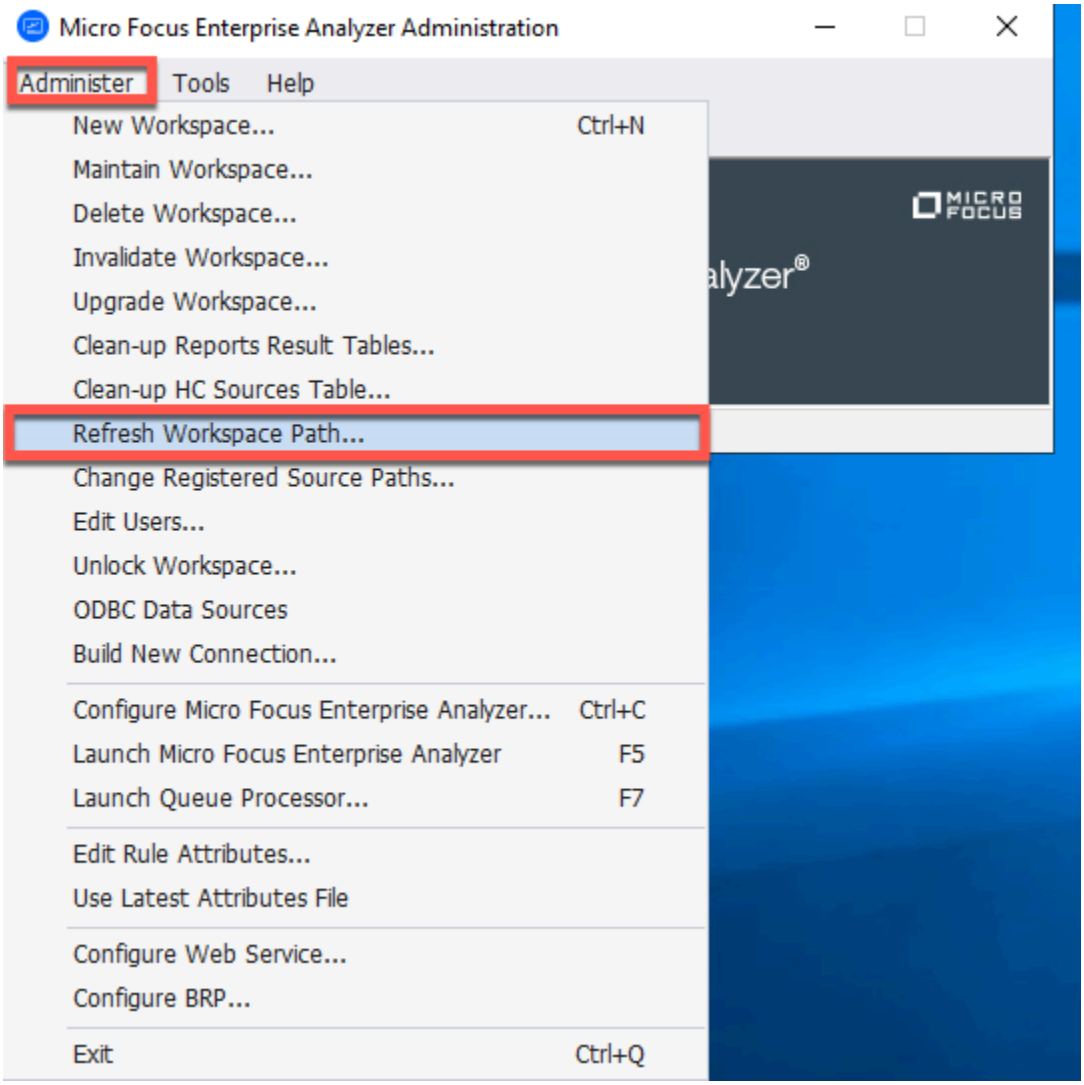
```
Cannot access the workspace directory D:\PhotonUser\My Files\Home Folder\EA_BankDemo.
 The workspace has been created on a non-shared disk of the EC2AMAZ-E6LC33H computer.
 Would you like to correct the workspace directory location?
```

Per risolvere questo problema, scegli OK per cancellare il messaggio, quindi completa i seguenti passaggi.

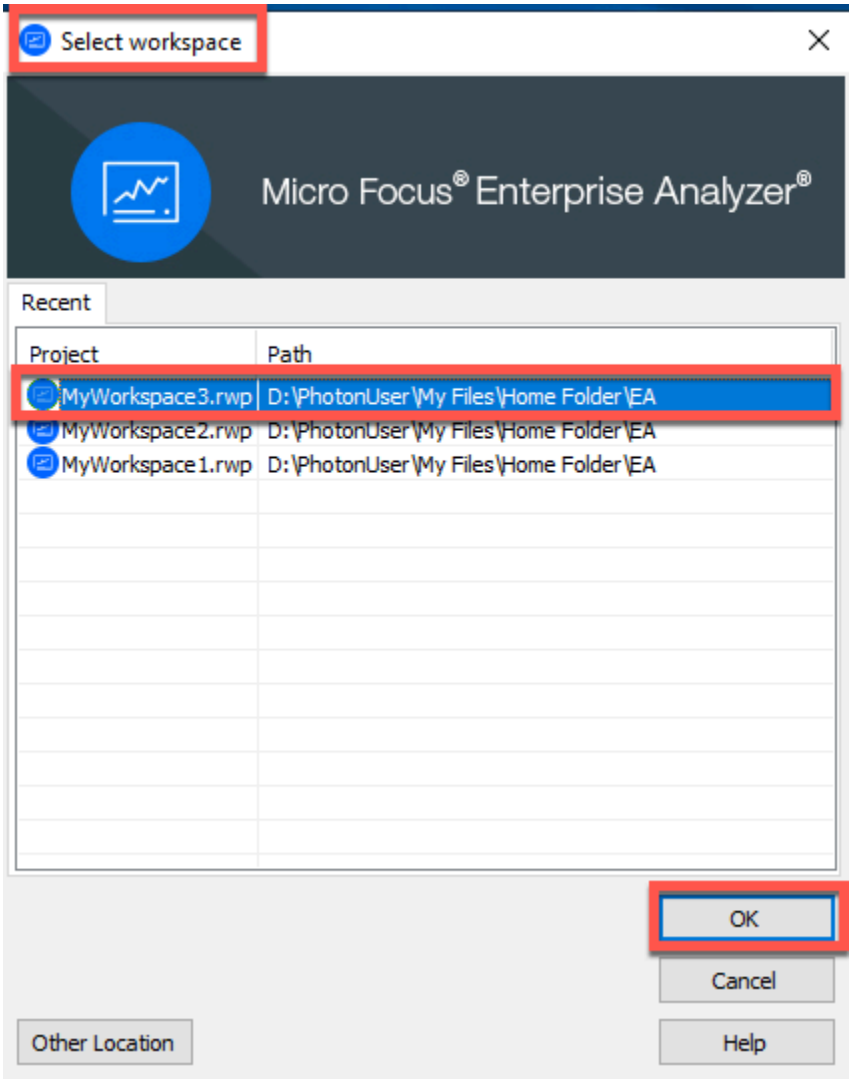
1. In WorkSpaces Applicazioni, scegliete l'icona Avvia applicazione sulla barra degli strumenti, quindi scegliete EA_Admin per avviare lo strumento di amministrazione di Enterprise Analyzer.



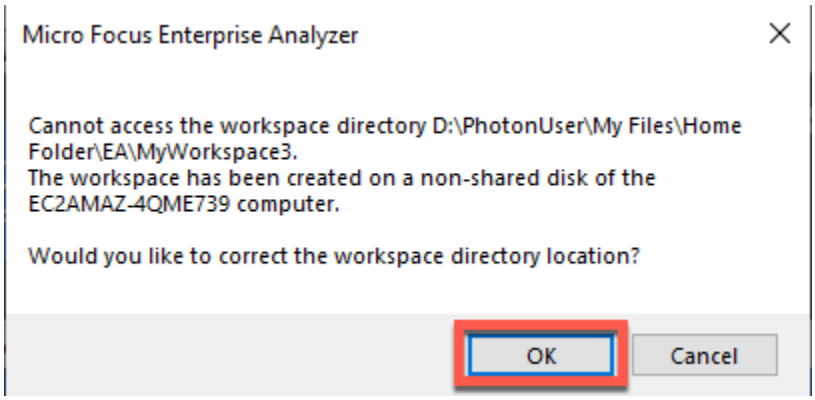
2. Dal menu Amministra, scegli Aggiorna percorso dell'area di lavoro... .



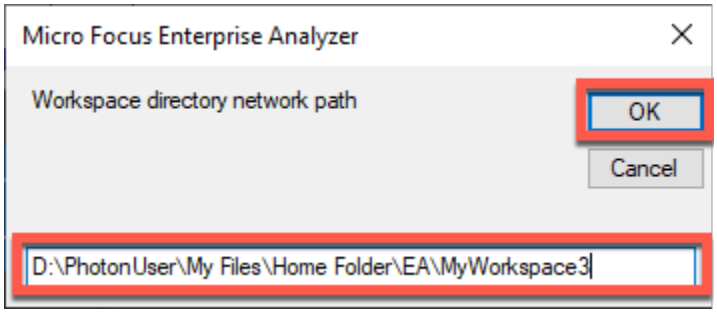
3. In Seleziona area di lavoro, scegli l'area di lavoro che desideri, quindi scegli OK.



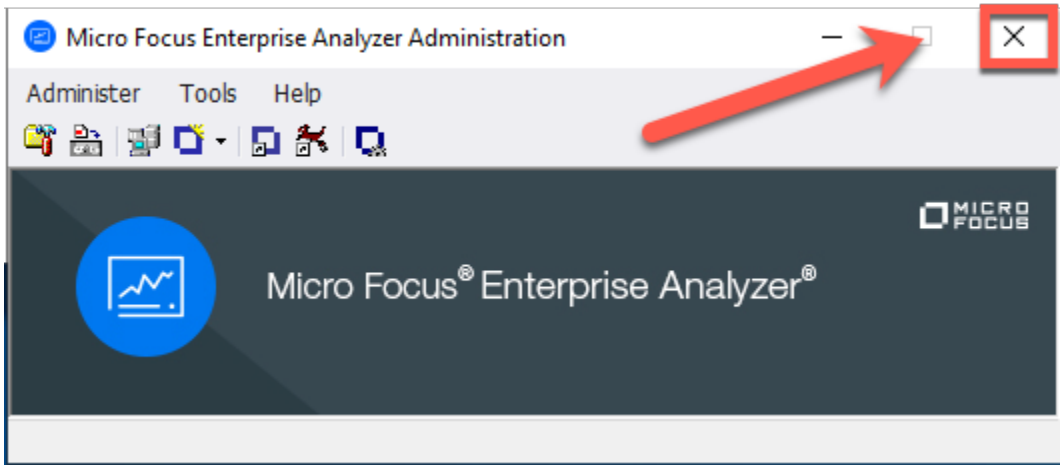
4. Scegli OK per confermare il messaggio di errore.



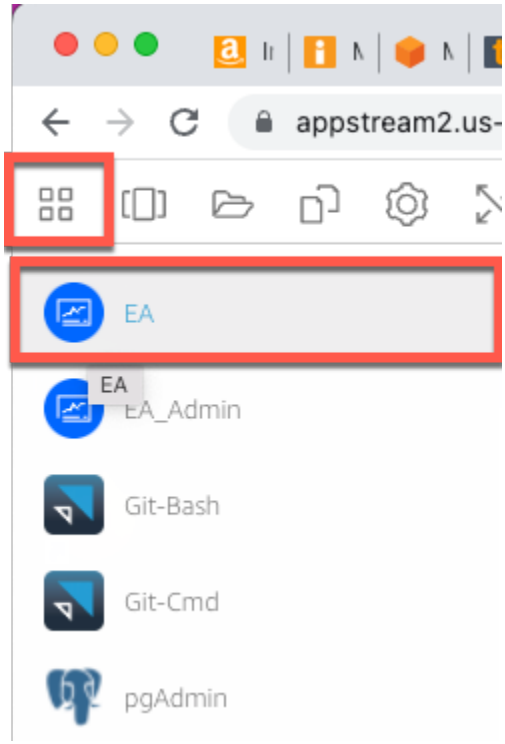
5. In Workspace Directory network path, inserite il percorso corretto del workspace, ad esempio. `D:\PhotonUser\My Files\Home Folder\EA\MyWorkspace3`



6. Chiudete lo strumento di amministrazione di Micro Focus Enterprise Analyzer.



7. In WorkSpaces Applicazioni, 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.

Eseguire la 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 Rocket Enterprise Developer sulle applicazioni WorkSpaces

Questo tutorial descrive come configurare Rocket Enterprise Developer (precedentemente 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 delle WorkSpaces applicazioni che AWS Mainframe Modernization condivide con il cliente e sulla creazione di flotte e stack di WorkSpaces applicazioni come descritto in. Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket Enterprise Developer

> **Important**
>
> I passaggi di questo tutorial presuppongono la configurazione WorkSpaces delle applicazioni utilizzando il modello scaricabile cfn-m2- .yaml CloudFormation . appstream-fleet-ea-ed Per ulteriori informazioni, consulta Tutorial: Configurazione WorkSpaces delle applicazioni da utilizzare con Rocket Enterprise Analyzer e Rocket 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 Rocket Software (precedentemente Micro Focus).

Contenuto dell'immagine

Oltre allo stesso Enterprise Developer, l'immagine contiene l'immagine contenente Rumba (un TN3270 emulatore). 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 sulle applicazioni WorkSpaces

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
- Eseguire la 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 CI/CD pipeline 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. CodeCommit 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/.
    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. Avvia una sessione con WorkSpaces Applications 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 disponibile 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 `amzn-s3-demo-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 alle WorkSpaces applicazioni 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:amzn-s3-demo-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` (alias`D: \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 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 in Gestione delle WorkSpaces applicazioni, 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 WorkSpaces Applications 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.

Eseguire la 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 l' CodeCommit archivio che hai creato per questo tutorial. Per ulteriori informazioni, consulta Eliminare un CodeCommit repository nella Guida per l'AWS CodeCommit utente.
- Elimina il database che hai creato per questo tutorial. Per ulteriori informazioni, consulta la sezione relativa all'eliminazione di un'istanza database.

Utilità batch disponibili in Mainframe AWS Modernization

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 dei dati nei database, lo scaricamento dei dati dai database e così DB2 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 di Rocket Software (precedentemente 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 batch M2SFTP
- Utilità batch M2WAIT
- TXT2Utilità PDF batch
- Utilità batch M2DFUTIL
- Utilità batch M2RUNCMD

Ubicazione binaria

Queste utilità sono preinstallate sui prodotti Rocket Enterprise Developer (ED) e Rocket Software (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 batch M2SFTP

M2SFTP è un programma di utilità JCL progettato per eseguire trasferimenti sicuri di file tra sistemi utilizzando 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 l'FTP del mainframe in JCLs modo che utilizzi SFTP, cambia 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 (comprese le applicazioni) WorkSpaces
- Esempio JCLs
- Riferimento ai comandi del client Putty SFTP (PSFTP)
- Fasi successive

Piattaforme supportate

È possibile utilizzare M2SFTP su una delle seguenti piattaforme:

- AWS Software Mainframe Modernization Rocket (precedentemente Micro Focus) gestito
- Rocket Software Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Rocket Software Enterprise Developer (ED) e Rocket Software 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, sarà necessario configurare M2SFTP come segue.

- Imposta le variabili di ambiente Rocket 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 DD`istruzione. Per ulteriori informazioni su queste variabili, consultate MFFTP Control Variables nella documentazione di Micro Focus.

Per testare la configurazione, consulta. Esempio JCLs

Configura M2SFTP per il runtime di modernizzazione AWS del mainframe su Amazon EC2 (comprese le applicazioni) WorkSpaces

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. Imposta le variabili di ambiente Rocket 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 DD`istruzione. Per ulteriori informazioni su queste variabili, consultate MFFTP Control Variables nella documentazione di Micro Focus.

Per testare la configurazione, consulta. Esempio JCLs

Esempio JCLs

Per testare l'installazione, è possibile utilizzare uno dei seguenti file JCL di esempio.

M2 .jcl SFTP1

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
/*
//*
//
```

M2 SFTP2 .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	Description
!	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 i 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	Description
mget	Scarica più file contemporaneamente
mkdir	Crea directory sul server remoto
Imput	Carica più file contemporaneamente
mv	Spostare o rinominare i file sul server remoto
aperto	Connect a un host
put	Carica un file dal tuo computer locale al server
pwd	Stampa la tua directory 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 i file sul server remoto
rmdir	Rimuove le directory sul server remoto

Fasi successive

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 SFTPGestione dei segreti AWS

Utilità batch M2WAIT

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 `C$SLEEP` fornito da Rocket Software (precedentemente Micro Focus) per attendere un periodo di tempo specificato.

> **Note**
>
> È possibile utilizzare gli alias Micro Focus per sostituire gli alias presenti negli script JCL. Per ulteriori informazioni, vedete JES Alias nella documentazione di Micro Focus.

Argomenti

- Piattaforme supportate
- Configura M2WAIT per AWS Mainframe Modernization Managed
- Configura M2WAIT for AWS Mainframe Modernization runtime su Amazon EC2 (incluse le applicazioni) WorkSpaces
- Esempio JCL

Piattaforme supportate

Puoi utilizzare M2WAIT su una delle seguenti piattaforme:

- AWS Software di modernizzazione del mainframe Rocket (precedentemente Micro Focus) gestito
- Rocket Software Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Rocket Software Enterprise Developer (ED) e Rocket Software Enterprise Server (ES).

Configura M2WAIT per 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 (incluse le applicazioni) WorkSpaces

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=*
//*
//
```

TXT2Utilità PDF 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 TXT2 PDF (z/OSfreeware). L'abbiamo modificata per eseguirla nell'ambiente di runtime del software AWS Mainframe Modernization Rocket (precedentemente Micro Focus).

Argomenti

- Piattaforme supportate
- Configura TXT2 PDF per la modernizzazione del AWS mainframe Managed
- Configura TXT2 PDF per il runtime di modernizzazione del AWS mainframe su Amazon EC2 (incluse le applicazioni) WorkSpaces
- Esempio di JCL
- Modifiche
- Riferimenti

Piattaforme supportate

È possibile utilizzare TXT2 PDF su una delle seguenti piattaforme:

- AWS Modernizzazione del mainframe: software Rocket gestito
- Rocket Software Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Rocket Enterprise Developer (ED) e Rocket Enterprise Server (ES).

Configura TXT2 PDF per la modernizzazione del AWS mainframe Managed

Se le applicazioni migrate sono in esecuzione su AWS Mainframe Modernization Managed, configura PDF come segue. TXT2

- Create una libreria REXX EXEC chiamata. `AWS.M2.REXX.EXEC` Scaricate questi moduli REXX e copiateli nella libreria.
  - `TXT2PDF.rex`- TXT2 PDF z/OS gratuito (modificato)
  - `TXT2PDFD.rex`- TXT2 PDF z/OS gratuito (non modificato)
  - `TXT2PDFX.rex`- TXT2 PDF z/OS gratuito (modificato)
  - `M2GETOS.rex`- Per verificare il tipo di sistema operativo (Windows o Linux)

Per testare la configurazione, consultaEsempio di JCL.

Configura TXT2 PDF per il runtime di modernizzazione del AWS mainframe su Amazon EC2 (incluse le applicazioni) WorkSpaces

Se le applicazioni migrate sono in esecuzione sul runtime di modernizzazione del AWS mainframe su Amazon EC2, configura il PDF come segue. TXT2

1. Imposta la variabile di ambiente Rocket Software sul valore appropriato, `MFREXX_CHARSET` ad esempio «" `A` per i dati ASCII.

   > **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_CHARSET`Ti `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`- TXT2 PDF z/OS gratuito (modificato)
   - `TXT2PDFD.rex`- TXT2 PDF z/OS gratuito (non modificato)
   - `TXT2PDFX.rex`- TXT2 PDF z/OS gratuito (modificato)
   - `M2GETOS.rex`- Per verificare il tipo di sistema operativo (Windows o Linux)

Per testare la configurazione, consultaEsempio 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 TXT2 conversione PDF.

```
//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 PDF. TXT2

```
//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 TXT2 PDF nell'ambiente di runtime del software AWS Mainframe Modernization Rocket, abbiamo apportato le seguenti modifiche:

- Modifiche al codice sorgente per garantire la compatibilità con il runtime REXX di Rocket Software
- 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

TXT2Riferimenti PDF e codice sorgente:

- Convertitore da testo a PDF
- z/OSStrumenti freeware TCP/IP e di posta
- TXT2Guida di riferimento per l'utente in PDF

Utilità batch M2DFUTIL

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 del 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
- Esempio JCLs

Piattaforme supportate

È possibile utilizzare M2DFUTIL su una delle seguenti piattaforme:

- Rocket Software (precedentemente Micro Focus) ES su Windows (64 bit e 32 bit)
- Rocket Software 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, CREDT, 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 regolari`compare.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 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 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 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 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 DSNAME 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

Esempio JCLs

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 batch M2RUNCMD

È possibile utilizzare M2RUNCMD, un programma di utilità batch, per eseguire comandi, script e chiamate di sistema di Rocket Software (precedentemente Micro Focus) direttamente da JCL invece di eseguirli da un terminale o da un prompt dei comandi. L'output dei comandi viene registrato nello spool log del processo batch.

Argomenti

- Piattaforme supportate
- Configura M2RUNCMD per il runtime di modernizzazione AWS del mainframe su Amazon EC2 (incluso 2.0) AppStream
- Esempio JCLs

Piattaforme supportate

È possibile utilizzare M2RUNCMD sulle seguenti piattaforme:

- Rocket Software Runtime (su Amazon EC2)
- Tutte le varianti dei prodotti Rocket Software Enterprise Developer (ED) e Rocket Software 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`

Esempio JCLs

Per testare l'installazione, è possibile utilizzare uno dei seguenti esempi JCLs.

RUNSCRL1.jcl

Questo esempio di JCL 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. Puoi anche scegliere di eseguire solo il secondo passaggio per eseguire i comandi di sistema e il software Rocket 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 ENVIRONMNET 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 SYS0UT 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  ************
```

RUNCMDL1.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 ENVIRONMNET VARIABLE
 echo $PATH
/*
//
```

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. Semplifica il processo di trasferimento dei set di dati dal mainframe al. Cloud AWS Le caratteristiche principali includono: individuazione dei set di dati e degli artefatti mainframe di origine e scalabilità ed efficienza per trasferimenti di dati più rapidi verso Amazon S3. File Transfer supporta vari tipi di set di dati mainframe come sequential, PDS, GDS, GDG e VSAM KSDS. Il servizio trasferisce i set di dati in un bucket Amazon S3 intermedio, li converte nella tabella codici di destinazione specificata e quindi li sposta nel bucket S3 di destinazione desiderato.

Argomenti

- Cos'è AWS Mainframe Modernization File Transfer?
- Installare un agente di trasferimento file
- Configurare un agente di trasferimento file
- Crea endpoint di trasferimento dati per File Transfer
- Crea attività di trasferimento in File Transfer
- Tutorial: Come iniziare a usare AWS Mainframe Modernization File Transfer
- Codifiche di origine e destinazione supportate in 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

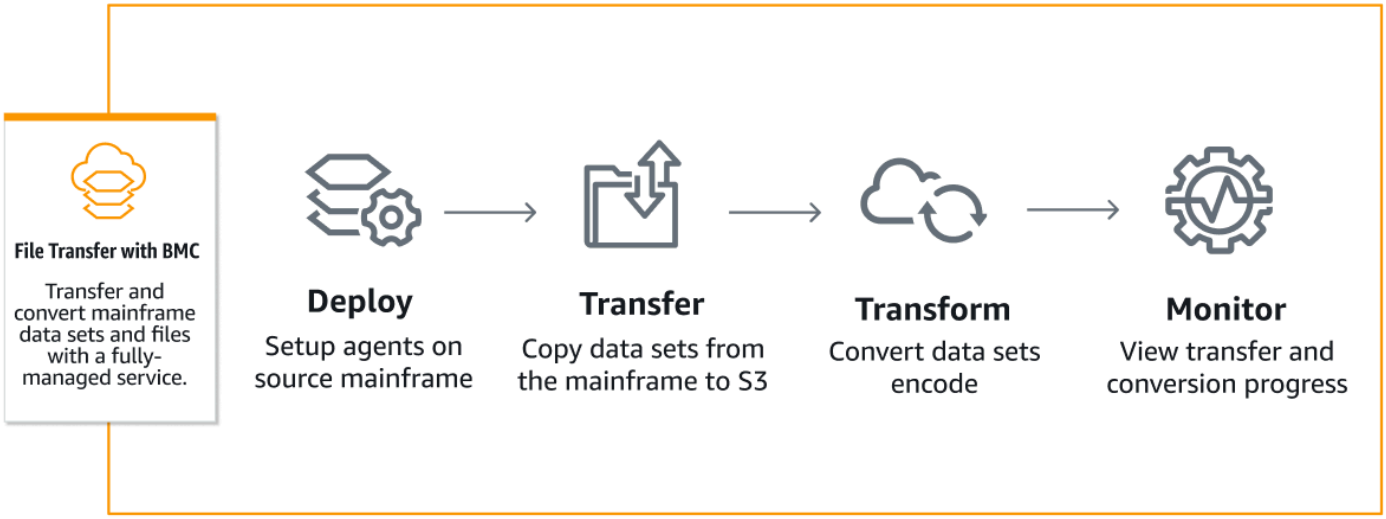
Vantaggi del trasferimento di file per la modernizzazione dei mainframe di AWS

AWS Mainframe Modernization File Transfer ti aiuta a trasferire set di dati dal mainframe ad Amazon S3. Alcuni vantaggi includono:

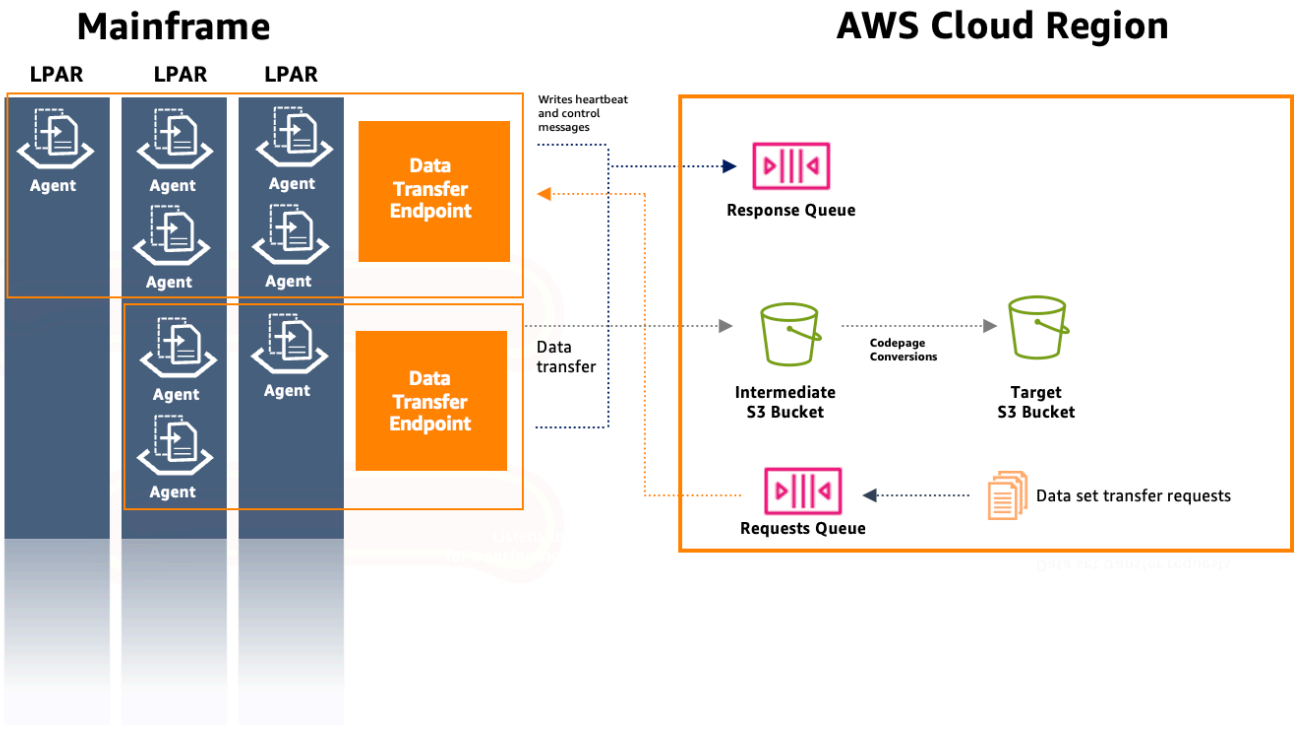
- Scoperta dei set di dati e degli artefatti mainframe di origine
- Trasferimenti automatici e conversione di set di dati
- Scalabilità, efficienza e velocità per ottenere trasferimenti di set di dati più rapidi verso AWS

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

È possibile utilizzare questo documento come step-by-step guida per installare un agente sul mainframe di origine.

> **Note**
>
> Questa guida è destinata solo ai programmatori di sistemi mainframe.

Argomenti

- Fase 1: Creare un set di dati ZfS per M2-Agent
- Fase 2: Formattare il set di dati come ZFs
- Fase 3: Montare il filesystem
- Fase 4: Verificare il montaggio
- Fase 5: Inserire OMVS
- Passaggio 6: Impostare la variabile di ambiente della directory di installazione dell'agente

- Fase 7: Impostare la variabile di ambiente della directory di lavoro
- Fase 8: Creare la cartella di lavoro
- Fase 9: Copiare il file tar dell'agente e copiare la directory di lavoro
- Fase 10: Completare l'installazione dell'agente

Fase 1: Creare un set di dati ZfS per M2-Agent

Create uno ZfS per l'installazione di M2-Agent utilizzando il seguente JCL (Job Control Language):

```
DEFINE   EXEC   PGM=IDCAMS
SYSPRINT DD     SYSOUT=A
SYSIN    DD     *
 DEFINE CLUSTER (NAME(yourhlq.M2AGENT.ZFS) -
 VOLUMES(*) -
 LINEAR CYL(1000 200))
```

Fase 2: Formattare il set di dati come ZFs

Dopo aver creato il set di dati, formattatelo come file system ZfS.

Un modo per farlo è usare il seguente JCL:

```
FORMAT  EXEC PGM=IOEAGFMT,
PARM=('-aggregate yourhlq.M2AGENT.ZFS -size 1200') ,
SYSPRINT DD  SYSOUT=*
```

Invia questo lavoro e controlla se è stato completato con successo.

Fase 3: Montare il filesystem

Per montare il filesystem, usa il comando. `MOUNT` È possibile montare il filesystem nella riga di comando in ISPF o in batch.

Esempio:

```
MOUNT FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/usr/lpp/aws/
m2-agent')
```

Utilizzerai questo punto di montaggio nel passaggio 6.

> **Note**
>
> La definizione del percorso di montaggio è facoltativa e a tale scopo è necessario utilizzare una directory esistente.

Fase 4: Verificare il montaggio

Verificate che il filesystem sia montato correttamente usando il `D OMVS,F` comando o controllando all'interno di Unix System Service (USS).

Fase 5: Inserire OMVS

Usa il seguente comando per inserire OMVS:

```
TSO OMVS
```

Passaggio 6: 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
```

> **Note**
>
> Il punto di montaggio è definito nel passaggio 3.

Fase 7: 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 8: Creare la cartella di lavoro

Utilizzate il seguente comando per impostare l'ambiente della directory di lavoro:

```
mkdir -p $WORK_DIR
```

Fase 9: Copiare il file tar dell'agente e copiare la directory di lavoro

Scarica il file tar dell'agente da AWS utilizzando il collegamento all'agente M2.

Il meccanismo di trasferimento dipenderà dal tuo ambiente, ma assicurati che il file tar venga trasferito in modalità binaria.

Fase 10: Completare l'installazione dell'agente

Segui questi passaggi per completare l'installazione dell'agente.

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-$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 il. `SYS2.AWS.M2 HLQ`

   Per creare l'agente File Transfer, aggiorna le tre variabili simboliche HLQ (High level qualifier) e utilizzale `VOLSER` successivamente in `AGNTPATH` JCL:

```
oedit $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.

Configurare un agente di trasferimento file

Dopo aver installato un agente di trasferimento file, segui questi passaggi per configurare l'agente. Se devi installare un nuovo agente, segui le istruzioni riportate nella the section called "Installa un agente di trasferimento file" pagina.

Argomenti

- Passaggio 1: Configurazione delle autorizzazioni e Started Task Control (STC)
- Fase 2: creazione di bucket Amazon S3
- Fase 3: Creare una chiave di crittografia gestita AWS KMS dal cliente
- Fase 4: Creare un Gestione dei segreti AWS segreto per le credenziali del mainframe
- Fase 5: Creare una policy IAM
- Fase 6: Creare un utente IAM con credenziali di accesso a lungo termine
- Fase 7: Creare un ruolo IAM che l'agente deve assumere
- Fase 8: Configurazione dell'agente

Passaggio 1: Configurazione delle autorizzazioni e Started Task Control (STC)

1. Aggiorna e invia uno dei seguenti `SYS2.AWS.M2.SAMPLIB(SEC#RACF)` (per la configurazione delle autorizzazioni RACF) o `SYS2.AWS.M2.SAMPLIB(SEC#TSS)` (per l'impostazione delle autorizzazioni TSS) in base alle relative istruzioni. Questi membri sono stati creati nel passaggio precedente. `CPY#PDS`

> **Note**
>
> `SYS2.AWS.M2`deve essere sostituito con il qualificatore di alto livello (HLQ) scelto durante l'installazione.

2. Aggiorna l'esportazione PWD in `SYS2.AWS.M2.SAMPLIB(M2AGENT)` STC JCL, se il percorso di directory predefinito dell'agente File Transfer () è stato modificato. `/usr/lpp/aws/m2-agent`
3. Aggiorna il PROC in base agli standard del tuo sito:
   a. Aggiorna la scheda PROC in base ai requisiti di installazione.
   b. Aggiorna STEPLIB con. `M2 LOADLIB PDSE ALIAS`
   c. Modifica PWD per indicare il percorso di installazione dell'agente (è incluso solo questo).
   d. Aggiorna `JAVA_HOME` se necessario.
4. Aggiorna e copia il `SYS2.AWS.M2.SAMPLIB(M2AGENT)` JCL in `SYS1.PROCLIB` uno dei file della tua PROCLIBs `PROCLIB` concatenazione.
5. Aggiungete `SYS2.AWS.M2.LOADLIB` all'elenco APF usando il seguente comando:

```
SETPROG APF ADD DSNAME(SYS2.AWS.M2.LOADLIB) SMS
```

6. Imposta il gruppo e il proprietario dell'agente sull'agente user/group (M2USER/M2GROUP). Usa il seguente comando in OMVS:

```
chown -R M2USER:M2GROUP $AGENT_DIR/current-version
```

> **Note**
>
> Modifica M2USER e M2GROUP con i nomi che hai usato nel processo delle definizioni di sicurezza.

Fase 2: creazione di bucket Amazon S3

AWS Mainframe Modernization File Transfer richiede un bucket Amazon S3 intermedio come area di lavoro. Ti consigliamo di creare un bucket specifico per questo scopo.

Facoltativamente, crea un nuovo bucket Amazon S3 di destinazione per i set di dati trasferiti. Altrimenti puoi anche usare il tuo bucket Amazon S3 esistente. Per ulteriori informazioni sulla creazione di bucket Amazon S3, consulta Creazione di un bucket.

Fase 3: Creare una chiave di crittografia gestita AWS KMS dal cliente

Per creare una chiave gestita dal cliente in AWS KMS

1. Apri la AWS KMS console all'indirizzohttps://console.aws.amazon.com/kms.
2. Scegli Customer managed keys nel riquadro di navigazione a sinistra.
3. Scegli Crea chiave.
4. In Configura chiave, scegli Tipo di chiave come Simmetrico e Utilizzo della chiave come Crittografa e decrittografa. Usa altre configurazioni predefinite.
5. Scegli Next (Successivo).
6. In Aggiungi etichette, aggiungi alias e descrizione per la tua chiave.
7. Scegli Next (Successivo).
8. In Definisci le autorizzazioni amministrative chiave, scegli almeno un utente e un ruolo IAM che amministra questa chiave.
9. Scegli Next (Successivo).
10. Facoltativamente, in Definisci le autorizzazioni amministrative chiave, scegli almeno un utente e un ruolo IAM che può utilizzare questa chiave.
11. Scegli Next (Successivo).
12. Nella sezione Modifica politica chiave, scegli Modifica e aggiungi la seguente sintassi alla politica chiave. Ciò consente al servizio AWS Mainframe Modernization di leggere e utilizzare queste chiavi per la crittografia/decrittografia.

> **Important**
>
> Aggiungere l'istruzione alle istruzioni esistenti. Non sostituire ciò che è già contenuto nella politica.

```
{
    "Sid" : "Enable AWS M2 File Transfer Permissions",
    "Effect" : "Allow",
```

```
    "Principal" : {
        "Service" : "m2.amazonaws.com"
    },
    "Action" : [
        "kms:Encrypt",
        "kms:Decrypt"
    ],
   "Resource" : "*"
},
```

13. Scegli Next (Successivo).
14. Nella pagina Revisione, controlla tutti i dettagli e scegli Fine.

Copia e salva l'ARN per la chiave gestita dal cliente aprendo la chiave KMS appena creata. Verrà utilizzato nella politica in un secondo momento.

Fase 4: Creare un Gestione dei segreti AWS segreto per le credenziali del mainframe

Le credenziali del mainframe sono necessarie per accedere ai set di dati da trasferire e queste devono essere archiviate come segrete. Gestione dei segreti AWS

Per creare un segreto Gestione dei segreti AWS

1. Apri la console di gestione di Secrets all'indirizzohttps://console.aws.amazon.com/secretsmanager.
2. Scegli Archivia un nuovo segreto.
3. In Scegli il tipo di segreto, scegli Altro tipo di segreto.
4. Utilizzate il valore chiave `userId` per l'UserID del mainframe che ha accesso ai set di dati. Utilizzate il valore chiave `password` per il campo della password.
5. Per la chiave di crittografia, scegli la chiave gestita AWS dal cliente creata in precedenza.
6. Scegli Next (Successivo).
7. Nella pagina Configura segreto, fornisci un nome e una descrizione.
8. Nella stessa pagina, modifica le autorizzazioni delle risorse e utilizza la seguente politica delle risorse in modo che il servizio di modernizzazione del AWS mainframe possa accedervi.

JSON

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

9. Scegli Salva per salvare le autorizzazioni aggiornate.
10. Scegli Next (Successivo).
11. Passa alla pagina Configura le rotazioni e scegli Avanti.
12. Nella pagina Revisione, controlla tutte le configurazioni e scegli Store per salvare il segreto.

> **Important**
>
> Le chiavi `password` segrete `userId` e le chiavi segrete fanno distinzione tra maiuscole e minuscole e devono essere immesse come mostrato.

Fase 5: Creare una policy IAM

Per creare una nuova policy con le autorizzazioni richieste per l'agente

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam.
2. Scegli Politiche in Gestione degli accessi.
3. Scegli Crea policy.
4. Nella pagina Specificare le autorizzazioni, in Policy editor, passa dall'editor visuale all'editor JSON e sostituisci il contenuto con il seguente modello:
5.

6. Sostituisci gli `111122223333` ARN presenti nella coda di richiesta e nella coda di risposta con il tuo account.

   > **Note**
   >
   > Si tratta di ARN wildcard che corrispondono alle due code Amazon SQS create durante l'inizializzazione dell'endpoint di trasferimento dati. Dopo aver creato un endpoint File Transfer, puoi facoltativamente sostituire questi ARN con i valori effettivi di Amazon SQS.

7. Sostituisci `file-transfer-endpoint-intermediate-bucket-arn` con l'ARN del bucket di trasferimento creato in precedenza. Lascia il carattere jolly «/*» alla fine.
8. Sostituisci `kms-key-arn` con l'ARN della AWS KMS chiave creata in precedenza.
9. Scegli Next (Successivo).
10. Nella pagina Rivedi e crea, aggiungi il nome e la descrizione della politica.
11. Scegli Crea policy.

Fase 6: Creare un utente IAM con credenziali di accesso a lungo termine

Crea un utente IAM che consenta all'agente mainframe di connettersi al tuo AWS account. L'agente si connetterà con questo utente e quindi assumerà un ruolo da te definito con le autorizzazioni per utilizzare le code di risposta e richiesta di Amazon SQS e per salvare i set di dati nei bucket Amazon S3.

Per creare questo utente IAM

1. Accedi alla console IAM all'indirizzohttps://console.aws.amazon.com/iam.
2. Scegli Utenti in Gestione degli accessi.
3. Selezionare Create user (Crea utente).
4. Aggiungi un nome utente significativo in Dettagli utente. Ad esempio, `Configure-ft-agent`.
5. Scegli Next (Successivo).
6. Nelle opzioni di autorizzazione, scegli l'opzione Allega direttamente le politiche ma non allegare alcuna politica di autorizzazione. Queste autorizzazioni saranno gestite da un ruolo che verrà allegato.
7. Scegli Next (Successivo).
8. Controlla i dettagli e scegli Crea utente.

9. Una volta creato l'utente, scegli l'utente e apri la scheda Credenziali di sicurezza.
10. In Chiavi di accesso, seleziona Crea chiave di accesso.
11. Quindi, scegli Altro quando richiesto per il caso d'uso.
12. Scegli Next (Successivo).
13. Facoltativamente, puoi impostare un tag descrittivo come,. `Access key for configuring file transfer agent`
14. Selezionare Create access key (Crea chiave di accesso).
15. Copia e salva in modo sicuro la chiave di accesso generata e la chiave di accesso segreta. Questi verranno utilizzati in seguito.

Per ulteriori informazioni sulla creazione della chiave di accesso IAM, consulta Gestione delle chiavi di accesso per gli utenti IAM.

> **Important**
>
> Salva la chiave di accesso e la chiave di accesso segreta visualizzate nell'ultima pagina della procedura guidata per la creazione delle chiavi di accesso, prima di scegliere Fine. Queste chiavi vengono utilizzate per configurare l'agente mainframe e non possono essere recuperate in seguito.

> **Note**
>
> Salva l'ARN dell'utente IAM utilizzato per impostare una relazione di trust con un ruolo IAM.

Fase 7: Creare un ruolo IAM che l'agente deve assumere

Per creare un nuovo ruolo IAM per l'agente

1. Scegli Ruoli nella console IAM all'indirizzohttps://console.aws.amazon.com/iam.
2. Scegli Crea ruolo.
3. Nella pagina Seleziona un'entità affidabile, scegli Criteri di attendibilità personalizzati per il tipo di entità affidabile.

4. Sostituisci la politica di fiducia personalizzata con la seguente e `<iam-user-arn>` sostituiscila con l'ARN dell'utente creato in precedenza.

   JSON

   ```
   {
       "Version":"2012-10-17",
       "Statement": [ {
            "Sid": "FileTransferAgent",
            "Effect": "Allow",
            "Principal": {
               "AWS": "<IAM-User-arn>"
            },
            "Action": "sts:AssumeRole"
       } ]
   }
   ```

5. Scegli Next (Successivo).
6. In Aggiungi autorizzazioni, filtra in base al nome della politica che hai creato in precedenza e selezionalo.
7. Scegli Next (Successivo).
8. Assegna un nome al ruolo e scegli Crea ruolo.

> **Note**
>
> Salva il nome del ruolo, che utilizzerai in seguito per configurare l'agente mainframe.

Fase 8: 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 `appication.properties` per aggiungere una configurazione degli ambienti utilizzando il seguente comando:

   ```
   oedit $AGENT_DIR/current-version/config/application.properties
   ```

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>
zos.complex-name=<File_Transfer_Endpoint_Name>
```

Dove:

- `AWS_ACCOUNT_ID`è l'ID dell' AWS account.
- `AWS_IAM_ROLE_NAME`è il nome del ruolo IAM creato inthe section called “Fase 7: Creare un ruolo IAM che l'agente deve assumere”.
- `AWS_IAM_ROLE_ACCESS_KEY`è la chiave di accesso dell'utente IAM creato inthe section called “Fase 6: Creare 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 inthe section called “Fase 6: Creare 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**
>
> È possibile fare in modo che l'agente File Transfer venga trasferito su più regioni e account AWS definendo più ambienti.

- (Facoltativo). `zos.complex-name`è il nome complesso che hai creato durante la creazione di un endpoint File Transfer.

> **Note**
>
> Questo campo è necessario solo se desideri personalizzare il nome complesso (che per impostazione predefinita è il nome sysplex) che è lo stesso che hai definito durante

la creazione dell'endpoint File Transfer. Per ulteriori informazioni, consulta the section called "Crea endpoint per il trasferimento dei dati".

> **Important**
>
> 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 configurare l'agente File Transfer per trasferire dati a più regioni e account AWS definendo le voci di ambiente.

   > **Note**
   >
   > Sostituite i valori con i valori dei parametri creati e configurati in precedenza.

   ```
   #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
```

Crea endpoint di trasferimento dati per File Transfer

Gli endpoint di trasferimento dati consentono la connettività con il mainframe di origine e supportano alta disponibilità, scalabilità e gestione semplificata degli agenti. I singoli agenti sono installati sul mainframe LPARs 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 quel trasferimento specifico. 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 per la modernizzazione del AWS mainframe Configurazione dell'agente di trasferimento file sul mainframe di origine.

Crea endpoint per il trasferimento dei dati

Per creare endpoint di trasferimento dati per File Transfer, segui 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/
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 sul mainframe di origine. 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 dell'endpoint di trasferimento dati.

      > **Note**
      >
      > Il nome dell'endpoint di trasferimento dati deve corrispondere al nome Sysplex, a meno che non si specifichi un nome complesso nella configurazione dell'agente.

   2. Descrizione facoltativa
   3. La chiave KMS utilizzata per crittografare il segreto.

      > **Note**
      >
      > È necessario aggiungere la seguente politica basata sulle risorse per KMS in modo che il servizio di modernizzazione del AWS mainframe possa leggere e utilizzare queste chiavi per la crittografia/decrittografia:

      ```
      {
        "Sid" : "Enable AWS M2 Permissions",
        "Effect" : "Allow",
        "Principal" : {
            "Service" : "m2.amazonaws.com"
                  },
         "Action" : [
             "kms:Encrypt",
             "kms:Decrypt"
                    ],
         "Resource" : "*"
      }
      ```

   4. 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 prima di essere convertiti e trasferiti nel bucket Amazon S3 di destinazione.

      .

> **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.

5. 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 endpoint per il trasferimento dei dati.

Verrai reindirizzato alla pagina di panoramica degli endpoint di trasferimento dati dove potrai visualizzare 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.

Crea attività di trasferimento in File Transfer

Le attività di trasferimento vengono utilizzate per specificare i set di dati da trasferire dal mainframe ad Amazon S3 e consentono di scegliere le opzioni di conversione della tabella codici.

Queste istruzioni presuppongono che tu abbia completato i passaggi indicati Configurazione per la modernizzazione del AWS mainframe e che tu abbia creato. the section called "Crea 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 in File Transfer, segui questi passaggi nella console di modernizzazione del AWS mainframe.

Per creare un'attività di trasferimento

> **Important**
>
> 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/
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, puoi scegliere qualsiasi endpoint di trasferimento dati per creare attività di trasferimento.
4. Nella pagina Crea attività di trasferimento, imposta le proprietà per l'operazione di trasferimento. Se non hai creato alcuna attività di trasferimento in precedenza, puoi creare la prima attività scegliendo l'opzione Crea attività di trasferimento.

   - In questa pagina, inserisci le informazioni di base dell'operazione di trasferimento, tra cui il nome, la descrizione e la chiave segreta dell'operazione di trasferimento.

     > **Note**
     >
     > - Crittografa il segreto utilizzando la chiave KMS definita con l'endpoint di trasferimento dati. Il segreto deve contenere le credenziali del mainframe necessarie per accedere ai set di dati sul mainframe utilizzando le chiavi and. `userId` `password` Per ulteriori informazioni, consulta il segreto di AWS Secrets Manager.
     > - 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.
     >
     >   JSON

```
{
  "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 per il trasferimento è 90 GB.

- Successivamente, seleziona la posizione del bucket Amazon S3 di destinazione in cui verranno trasferiti i set di dati di destinazione dal mainframe.
- Verrà selezionato l'endpoint di trasferimento dati scelto in precedenza. È inoltre possibile selezionare un altro endpoint tra gli endpoint disponibili.

5. Scegli Next (Successivo).
6. Nella pagina Aggiungi set di dati, nella sezione Configurazione dell'attività di trasferimento, puoi scegliere di configurare l'attività di trasferimento in modalità binaria o di convertire e trasferire i set di dati.

   - L'opzione Trasferimento in modalità binaria consente di trasferire i set di dati ignorando le conversioni delle tabelle codici e conservando i relativi byte RDW (Record Descriptor Word).
   - L'opzione Trasferisci e converti i set di dati consente di trasferire i set di dati impostando le tabelle codici di origine e di destinazione per i set di dati. Puoi vedere le tabelle codici disponibili per File Transfer nella the section called "Pagine di codice sorgente e di destinazione supportate" pagina.
7. Inserisci la tua query nel mainframe Search for data sets per cercare nel mainframe i set di dati da includere nell'operazione di trasferimento. Scegli Visualizza set di dati.

   I seguenti simboli jolly possono essere utilizzati come parte dei criteri di ricerca dei set di dati per il mainframe:

   - Un asterisco singolo (*) come qualificatore (tra i periodi o dopo l'ultimo periodo) corrisponde a un singolo qualificatore in quella posizione.

- Un singolo asterisco (*) all'interno di un qualificatore corrisponde a zero o più caratteri in quella posizione.
- Un doppio asterisco (**) come qualificatore (tra i periodi o dopo l'ultimo periodo) corrisponde a zero o più qualificatori in quella posizione.
- Un doppio asterisco (**) all'interno di un qualificatore non è un'interrogazione valida.
- Un singolo segno di percentuale (%) corrisponde a qualsiasi carattere alfanumerico o nazionale presente in quella posizione. Puoi utilizzare fino all'otto percento in ogni qualificatore.

> **Note**
>
> Ti consigliamo di terminare sempre i criteri di ricerca con un punto seguito da un doppio asterisco (.**) e di affinare ulteriormente la ricerca, se necessario.

Per ulteriori informazioni sulle regole jolly, consulta la sezione Filtraggio dei nomi dei set di dati nella documentazione IBM.

8. Questi set di dati verranno caricati nella sezione Set di dati Mainframe, dove puoi cercare o scegliere uno o più set di dati per i quali configurare le conversioni della tabella codici. Questi set di dati scelti verranno visualizzati nella sezione Set di dati aggiunti. Se non viene caricato alcun set di dati, è necessario rivisitare il passaggio 7.

> **Note**
>
> È possibile selezionare set di dati da più query di ricerca e aggiungerli all'attività di trasferimento.

9. Nella sezione Set di dati aggiunti, vedrai il nome, il tipo e il nome del volume dei tuoi set di dati.

> **Important**
>
> Per l'opzione Trasferisci e converti i set di dati, devi inserire manualmente la tabella codici sorgente e la tabella codici di destinazione per ciascuno dei set di dati scelti. La tabella codici sorgente è il formato del set di dati di origine e la tabella codici di

destinazione è il formato del set di dati di destinazione utilizzato per convertire i set di dati e archiviarli nel bucket Amazon S3 di destinazione.

10. Dopo aver confermato i set di dati nella sezione Set di dati aggiunti (e le tabelle codici sorgente e di destinazione per l'opzione Trasferisci e converti i set di dati), scegli Avanti.
11. Nella pagina Rivedi e crea, puoi rivedere o modificare le informazioni relative all'attività di trasferimento.
12. Quindi, scegli Crea attività di trasferimento.

> **Important**
>
> Scegliendo il pulsante Crea attività di trasferimento verrà avviato il trasferimento dei dati, fatturabile in base alla pagina dei prezzi di AWS Mainframe Modernization. Questa fatturazione si basa sulla quantità di dati (GB) trasferiti misurata dalla dimensione del set di dati.

Visualizza le attività di trasferimento

Per visualizzare le attività di trasferimento in 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/
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, seleziona l'endpoint di trasferimento dati per visualizzare le attività di trasferimento.
4. Per gli endpoint con attività di trasferimento preesistenti, queste verranno visualizzate 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 di

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 "Configurare 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 "Crea 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 "Crea attività di trasferimento" 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 sezionethe section called "Visualizza le attività di trasferimento".

Codifiche di origine e destinazione supportate in AWS Mainframe Modernization File Transfer

AWS Mainframe Modernization File Transfer supporta vari tipi di set di dati e opzioni di conversione della tabella codici.

Tipi di set di dati mainframe

AWS Mainframe Modernization File Transfer supporta i seguenti tipi di set di dati mainframe:

- Non VSAM: sequenziale (PS), PDS, GDS, GDG
- Tipi di VSAM: KSDS

Pagine codici supportate

AWS Mainframe Modernization File Transfer supporta le seguenti code page per la conversione di set di dati (da/a):

"BIG5", "BIG5_HKSCS», «CESU_8", «EUC_JP», «EUC_KR», "", "", «GBK», GB18030 "GB2312", "", ",", ",IBM00858", ",IBM01140", ",IBM01141" ","IBM01142, "IBM01143", ",IBM01144", ",IBM01145", ",IBM01146", ",IBM01147", ",IBM01148", ",IBM01149", ",IBM037"», IBM1026 ",", IBM1047 ",IBM273"», ",IBM277", "», IBM278 ",IBM280"», ",IBM284", IBM285 "», "», IBM290 "", IBM297 "", IBM420 "", "IBM424", "IBM437", "", IBM500 "", IBM775 "",» IBM850"," IBM852 "," ",IBM855" ",IBM857" "," IBM860 ",", IBM861 "", IBM862 "," ",IBM863", "IBM864", "IBM865", "", IBM866 "", IBM868 "", "IBM869IBM870", «IBM_THAI», «ISO_2022_CN», «ISO_2022_JP», «ISO_2022_JP_2", «ISO_2022_IT», «ISO_2022_IT», «ISO_2022_IT», «ISO_8859_1", «ISO_8859_13", «ISO_8859_15", «ISO_8859_16", «ISO_8859_2", «ISO_8859_3", «ISO_8859_4", «ISO_8859_5", «ISO_8859_6", «ISO_8859_7", «ISO_8859_7", «ISO_8859_7" 8", «ISO_8859_9", "JIS_X0201", «JIS_X0212_1990",» IBM871 IBM918 KOI8_R», "KOI8_U», «SHIFT_JIS», «TIS_620", «US_ASCII», «UTF_16", «UTF_16BE», «UTF_16LE», «UTF_32", «UTF_32BE», «UTF_32LE», «UTF_8", «WINDOWS_1250", «WINDOWS_1250", «WINDOWS_1250", «WINDOWS_12_12" 51", «WINDOWS_1252", «WINDOWS_1253", "WINDOWS_1254", «WINDOWS_1255", «WINDOWS_1256", «WINDOWS_1257", «WINDOWS_1258", «WINDOWS_31J», «BIG5X__HKSCS_2001", «BIG5X__SOLARIS», «X_EUCJPOPEN», «X_EUC_JP_LINUX», «X_EUC_TW», «X_", «X_", «X_IBM1006", «X_IBM1025", «X_", «X_IBM1046", «X_IBM1097» IBM1098 IBM1112 IBM1122 IBM1123", «X_IBM1124", «X_IBM1129", «X_", «X_IBM1166", «X_IBM1364", «X_IBM1381», «X_ IBM29626 C», «X_IBM1383", «X_", «X_IBM300", «X_IBM33722", «X_", «X_IBM737", «X_IBM833", «X_IBM834", «X_", «X_IBM856", «X_IBM874", «X_IBM875», «X_IBM921», «X_», «X_IBM922», «X_IBM930», «X_IBM933», «X_», «X_IBM935», «X_IBM937», «X_» _ IBM939 ", «X_IBM942", «X_", «X_IBM943", «X_ IBM942 C», «X_", «X_ IBM943 C», «X_ IBM948IBM949", «X_", «X_ IBM949 C», «X_IBM950", «X_IBM964", «X_IBM970", «X_ISCII91", «X_ISO_2022_CN_CNS», «X_ISO_2022_CN_GB», «X_ISO_8859_11", «X_JIS0208", «X_JISAUTODETECT», «X_JOHAB», «X_MACARABIC», «X_MACARABIC», «X_MACEUROPA CENTRALE», «X_MACCROATIAN», «X_MACCYRILLIC», «X_MACDINGBAT», «X_MACGREEK», «X_MACHEBREW», «X_MACICELAND», «X_MACROMANIA», «X_MACSYMBOL», «X_MACTHAI», «X_MACTURKISH», «X_MACTURKISH», «X_MACUKRAINE»», «MS932X__0213", «MS950X__HKSCS», «MS950X__HKSCS_XP», «X_MSWIN_936", «X_PCK», «X_SJIS_0213", «X_UTF_16LE_BOM», «X_UTF_32 BE_BOM», «X_UTF_32LE_BOM», «X_WINDOWS_50220", «X_WINDOWS_50221", «X_WINDOWS_874", «X_WINDOWS_949", «X_WINDOWS_950", «X_WINDOWS_ISO2022j»

AWS Transform per modernizzare le applicazioni mainframe

AWS Transform for mainframe consente di modernizzare più rapidamente le applicazioni mainframe COBOL esistenti in applicazioni Java automatizzando le complesse e dispendiose attività di analisi delle codebase, pianificazione della trasformazione, generazione di documentazione tecnica, estrazione della logica aziendale, scomposizione del codice, pianificazione delle ondate ed esecuzione del refactoring e del reforging. Riduce i costi e la complessità della modernizzazione del mainframe sfruttando l'intelligenza artificiale generativa e l'automazione, preservando al contempo la logica di business mission critical. AWS Transform l'interfaccia in linguaggio naturale e l'approccio orientato agli obiettivi vi consentono di mantenere il controllo della trasformazione, consentendovi di concentrarvi sulle priorità strategiche, mentre l'automazione si occupa della parte pesante del percorso di modernizzazione.

Per ulteriori informazioni su funzionalità e caratteristiche chiave, procedure dettagliate di alto livello e coinvolgimento umano negli input e nell'elaborazione, consulta Migrazione e modernizzazione delle applicazioni mainframe nella Guida per l'utente.AWS Transform

Vantaggi principali

AWS Transform le funzionalità di modernizzazione delle applicazioni mainframe offrono molteplici vantaggi. Alcuni dei quali includono:

- Accelera il percorso di modernizzazione del mainframe con l'intelligenza artificiale generativa: ti AWS Transform aiuta a trasformare il tuo codice COBOL in codice Java moderno entro pochi mesi, anziché la cronologia tradizionale in anni.
- Colma le lacune nelle conoscenze: AWS Transform genera una documentazione completa per le applicazioni mainframe, colmando il divario di conoscenze e consentendo decisioni più informate.
- Preservate la logica aziendale fondamentale: AWS Transform preserva la logica aziendale fondamentale dei vostri sistemi legacy rifattorizzandoli in applicazioni Java moderne e ottimizzate per il cloud.
- Scomposizione dei domini logici aziendali e tecnici: scompone AWS Transform automaticamente la codebase del mainframe in domini aziendali distinti per ridurre lo sforzo manuale e il tempo necessari per l'analisi e la scomposizione del codebase.
- Funzionalità Human in the loop (HITL): AWS Transform forniscono un approccio autonomo e orientato agli obiettivi che consente di mantenere il controllo del percorso di modernizzazione del mainframe.

Procedura dettagliata sulla trasformazione della console delle applicazioni mainframe

Nell'esperienza AWS Transform web, è possibile eseguire la trasformazione delle applicazioni mainframe da COBOL a Java. Per capire come utilizzare questa funzione, seguite tutti i passaggi indicati nella pagina Trasformazione delle applicazioni mainframe della Guida per l'AWS Transform utente.

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 accedervi e utilizzarle. AMIs

Prerequisiti

- Assicurati di avere accesso da amministratore a un AWS account in cui 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.

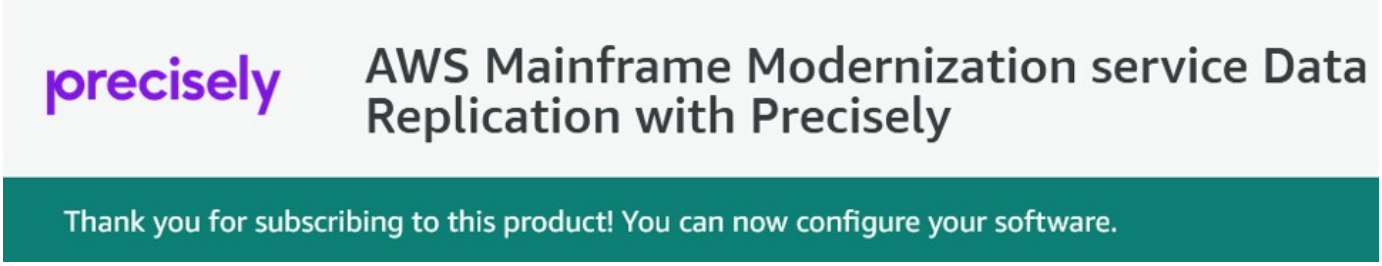
Iscriviti all'Amazon Machine Image

Quando ti abboni a un prodotto AWS Marketplace, puoi avviare un'istanza dall'AMI del prodotto.

1. Accedi Console di gestione AWS e apri la Marketplace AWS console all'indirizzo https://console.aws.amazon.com/marketplace.
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).
3. Accedi a uno dei seguenti link in base al tuo caso d'uso:

- Replica dei dati per IBM z/OS: https://aws.amazon.com/marketplace/pp/prodview-doe2lroefogia
- Replica dei dati per IBM i: https://aws.amazon.com/marketplace/pp/prodview-iqrkflccxf7ko

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.



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 su /marketplace. https://console.aws.amazon.com
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 Guida per l'utente di Amazon EC2.
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.

   JSON

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

```
    "Statement": [
        {
            "Action": ["aws-marketplace:MeterUsage"],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Creazione di un ruolo IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello 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. Scegli Next (Successivo).
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. Scegli Next (Successivo).
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, scegliere Instances (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.

Per ulteriori informazioni su come iniziare a utilizzare la replica AWS dei dati per IBM i, consulta Panoramica sulla replica dei dati.

Modernizzazione del mainframe AWS Conversione di codice con mLogica

Modernizzazione del mainframe AWS La conversione del codice con mLogica (conversione del codice) è una Modernizzazione del mainframe AWS funzionalità che converte automaticamente il codice di Assembler del z/OS mainframe in COBOL. È possibile utilizzare la conversione del codice per estrarre un'immagine assembler utilizzando il AWS CodeBuild servizio per la conversione del codice previsto con il... Account AWS

Argomenti

- Cos'è la conversione di Assembler con mLogica?
- Comprendi la fatturazione della conversione del codice per la conversione in Assembler
- Concetti di conversione del codice
- Comprendi i componenti e i processi per la conversione del codice
- Tutorial: converti il codice da Assembler a COBOL in Modernizzazione del mainframe AWS

Cos'è la conversione di Assembler con mLogica?

Modernizzazione del mainframe AWS La conversione del codice con mLogica (conversione del codice) converte z/OS automaticamente il codice di Assembler del mainframe in COBOL. Il servizio viene eseguito all'interno dell'utente Account AWS e non trasmette o archivia il codice sorgente di Assembler o COBOL all'esterno di. Account AWS La conversione del codice consente all'account autorizzato di estrarre un'immagine Assembler utilizzando il AWS CodeBuild servizio per la conversione del codice desiderata.

Modernizzazione del mainframe AWS offre la possibilità di configurare build e pipeline di integration/continuous distribuzione continua (CI/CD) per le applicazioni migrate. Queste build e pipeline utilizzano AWS CodeBuild Amazon S3 per fornire questa funzionalità. AWS CodeBuild è un servizio di compilazione completamente gestito che compila il codice sorgente, esegue test unitari e produce artefatti pronti per la distribuzione. Amazon S3 è un servizio di storage di oggetti che offre scalabilità, disponibilità dei dati, sicurezza e prestazioni all'avanguardia nel settore.

Argomenti

- Compilatori di conversione del codice

- Architettura di conversione del codice
- Approccio all'automazione
- Sicurezza
- Risorse aggiuntive

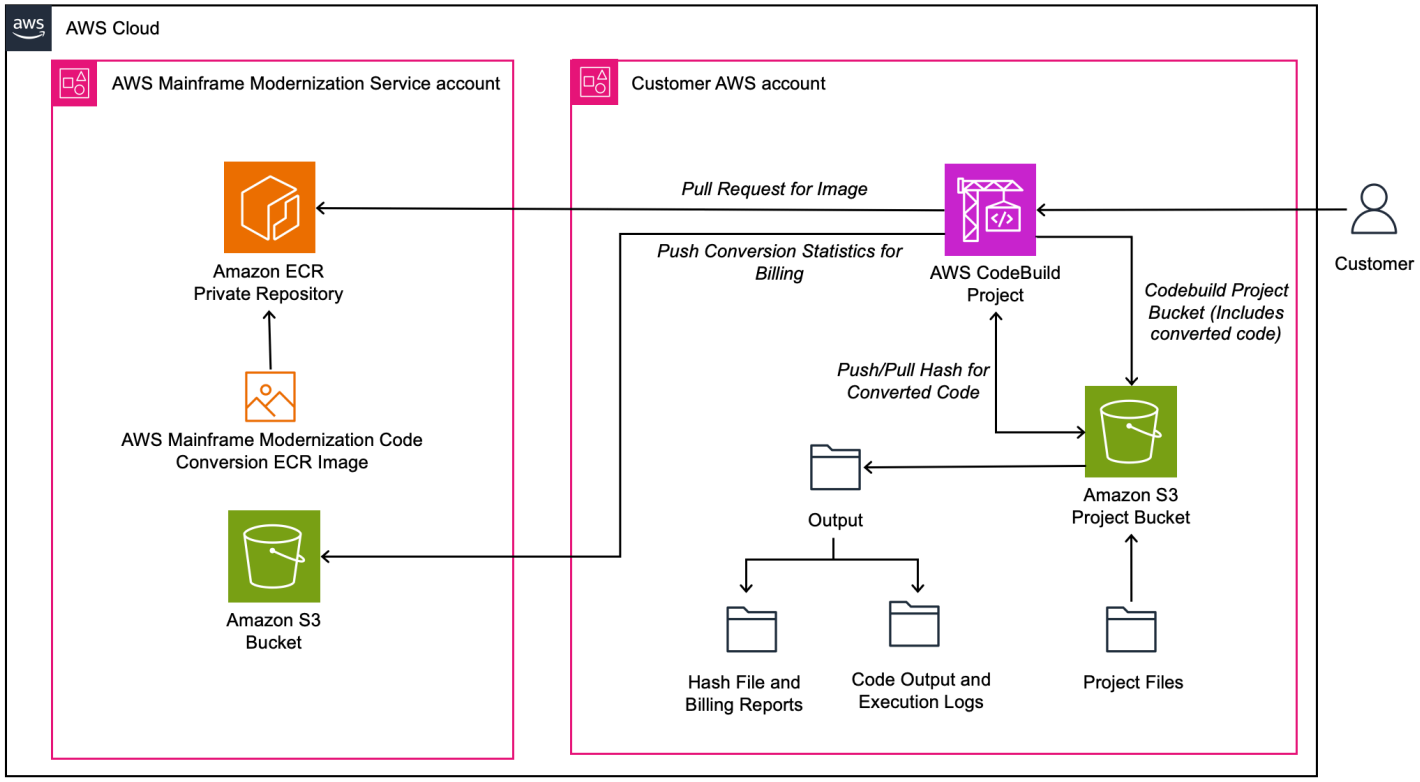
Compilatori di conversione del codice

La conversione del codice può essere configurata per emettere COBOL, adatto alla compilazione e all'esecuzione in diversi ambienti di destinazione con compilatori diversi. Alcuni di questi includono:

- Ripiattaforma di M2 con Rocket Software (precedentemente Micro Focus) e altri ambienti Rocket Enterprise Server
- Ripiattaforma M2 con NTT DATA Enterprise COBOL () UniKix
- MLogica LIBER*COBOL
- z/OSMainframe che utilizza IBM Enterprise COBOL
- Molto iCOBOL

Architettura di conversione del codice

Di seguito è riportato un diagramma architettonico per il processo di conversione del codice:



Approccio all'automazione

Per utilizzare Code conversion with CodeBuild, il codice Assembler deve essere caricato in un bucket Amazon S3, per configurare successivamente i parametri di conversione e richiamare CodeBuild un progetto per eseguire ogni fase del processo di conversione. Il codice COBOL di destinazione viene automaticamente archiviato in un percorso specificato nel bucket Amazon S3.

Sicurezza

Modernizzazione del mainframe AWS La conversione del codice consente la conversione mantenendo tutto il codice sorgente e di destinazione nel tuo. Account AWS Il codice Assembler di origine, il codice COBOL di destinazione e i file di configurazione vengono archiviati nel bucket Amazon S3. Lo strumento di conversione automatizzato funziona come un contenitore nell'ambiente del CodeBuild tuo. Account AWS Il codice rimane sempre nel tuo account.

Per consentire allo strumento di conversione di accedere al tuo bucket Amazon S3, concedi le autorizzazioni al bucket a un ruolo. Servizio AWS Durante la configurazione CodeBuild, imposterai questo ruolo di servizio in modo da CodeBuild poter accedere all'immagine del contenitore e accedere al tuo bucket Amazon S3.

Risorse aggiuntive

Oltre athe section called "Tutorial: converti il codice da Assembler a COBOL", ecco alcune risorse aggiuntive in cui puoi imparare a creare i CloudFormation modelli e altre informazioni sulla conversione di Assembler in COBOL.

- Link al workshop per la conversione automatizzata del codice da Assembler a COBOL:. https://catalog.workshops.aws/awsm2ccm-assembler-cobol/en-US
- Post sul blog:. https://aws.amazon.com/blogs/migration-and-modernization/unlocking-new-potential-transform-your-assembler-programs-to-cobol-with-aws-mainframe-modernization/

Comprendi la fatturazione della conversione del codice per la conversione in Assembler

Consulterai questa pagina per comprendere l'ambito e la procedura di fatturazione per la conversione del codice prima di effettuare la conversione effettiva. La sezione relativa al calcolo della fatturazione menziona il processo attraverso il quale la conversione da Assembler a COBOL viene addebitata per ogni riga di codice.

Conversione, fatturazione e ambito del codice.

La conversione del codice Assembler genera addebiti (rapporti di fatturazione) Account AWS solo dopo aver completato la fase di conversione. L'addebito si basa sul numero di righe di codice convertite. Se si eseguono più passaggi di conversione, ad esempio dopo aver aggiunto un nuovo codice Assembler, modificato la configurazione di conversione o applicato una nuova versione del contenitore, per calcolare l'addebito vengono utilizzate solo le righe modificate e/o le nuove righe aggiunte. Non ti addebiteremo due volte per la conversione della stessa riga di codice nello stesso programma.

> **Note**
>
> I moduli con righe di codice modificate e tutte le righe di codice in programmi nuovi o rinominati verranno addebitati.

Per evitare addebiti multipli, Code conversion memorizza un file binario codificato per ogni modulo Assembler o Macro nel bucket del progetto. `<Project_bucket>/awsm2ccm-do-not-delete/<AWS_account_number>/Hash` Questi file codificati non contengono alcun codice cliente.

> **Important**
>
> Non modificate o eliminate manualmente questi file. Le modifiche possono comportare più fatturazioni per la conversione degli stessi componenti.

Il rapporto di analisi della conversione del Modernizzazione del mainframe AWS codice («Rapporto di analisi») fornisce ai clienti dettagli sull'ambito di conversione previsto, sul risultato e sulla fatturazione per garantire aspettative accurate sulla conversione effettiva. La conversione può comportare la mancata conversione di alcune righe di codice, la conversione parziale di alcune righe di codice e la conversione completa di alcune righe di codice. Il rapporto di analisi mostra il numero di righe di codice per ogni categoria. È necessario eseguire e leggere il rapporto di analisi prima di elaborare qualsiasi conversione di programmi, macro e quaderni. Una volta che il cliente esamina il rapporto di analisi e concorda con l'ambito riportato, il risultato previsto e la fatturazione prevista, può procedere con l'esecuzione della conversione.

> **Note**
>
> Eseguendo il **`Convert`** comando AWS Mainframe Modernization Code Conversion, confermi di aver eseguito e letto il report di analisi e concordi con il risultato previsto e il numero fatturabile di righe di codice.

Ambito della conversione

Modernizzazione del mainframe AWS La conversione del codice elabora tutte le righe di codice di tutti i componenti assembler, macro e copybook disponibili nelle directory scrlib e macrolib nella posizione di origine S3 configurata. I programmi Assembler e tutte le macro e i quaderni a cui si fa riferimento in un programma Assembler rientrano nell'ambito di applicazione. I componenti di macro e quaderni a cui non fa riferimento un programma assembler sono considerati fuori ambito e non convertiti. Durante l'elaborazione, il convertitore esegue algoritmi avanzati che considerano ogni componente dell'ambito in modo olistico. Tutte le righe di codice di questi componenti partecipano all'elaborazione indipendentemente dal fatto che siano totalmente convertite, parzialmente convertite o meno. Modernizzazione del mainframe AWS La conversione del codice ignora le righe vuote e non

le conta come righe di codice. Le righe di commento e le righe contenenti qualsiasi altro testo (ad esempio, le istruzioni JCL per l'assembler incorporate in JCL) vengono conteggiate come righe di codice per la fatturazione.

Calcolo della fatturazione

Modernizzazione del mainframe AWS Costi di conversione del codice per i componenti inclusi nell'ambito di applicazione nella loro interezza. Ciò significa che addebita ogni riga di codice all'interno di ciascun componente incluso nell'ambito, comprese le righe che non è stato possibile convertire, che sono state convertite parzialmente e sono state convertite completamente. Modernizzazione del mainframe AWS La conversione del codice somma tutte le righe di codice dei componenti forniti per l'elaborazione (inclusi i programmi assembler, i quaderni con riferimento e le macro con riferimento) e utilizza il numero totale di righe di codice per la fatturazione.

> **Note**
>
> I quaderni e le macro a cui non fa riferimento un programma Assembler non sono considerati validi.

Ad esempio, supponiamo che un programma abbia 1.000 righe di codice:

- 700 righe vengono completamente convertite
- 200 righe vengono parzialmente convertite
- 100 righe non vengono convertite

1.000 righe di codice verrebbero elaborate e saranno fatturabili.

Migliorare la conversione

Se come cliente desideri un tasso di conversione più elevato per le righe di codice o hai altri requisiti specifici, puoi contattare i AWS rappresentanti per ulteriori opzioni di coinvolgimento, come un intervento di calibrazione o l'assistenza di servizi professionali.

Concetti di conversione del codice

Per scoprire come avviene la conversione del codice, è importante comprendere alcuni concetti chiave come la gestione delle macro, le tabelle codici e. CodeBuild

Argomenti

- Gestione delle macro
- Pagine codici (EBCDIC o ASCII)
- CodeBuild

Gestione delle macro

Il codice Mainframe Assembler utilizza spesso le macro per incapsulare funzionalità da riutilizzare. Il comportamento delle macro viene in genere determinato in fase di esecuzione dell'applicazione in base ai parametri trasmessi da un programma Assembler. La conversione del codice fornisce diversi meccanismi per espandere le macro di Assembler prima della conversione in COBOL.

Pagine codici (EBCDIC o ASCII)

Mainframe Assembler contiene spesso caratteri letterali espressi come valori esadecimali corrispondenti ai caratteri EBCDIC. La conversione del codice offre una funzionalità configurabile per gestire automaticamente i caratteri letterali in ASCII quando si emette COBOL per ambienti ASCII.

CodeBuild

La conversione del codice è disponibile tramite il servizio. AWS CodeBuild AWS CodeBuild è uno strumento di automazione della compilazione originariamente progettato come parte di una CI/CD pipeline. In Modernizzazione del mainframe AWS, AWS CodeBuild viene utilizzato per automatizzare lo strumento di conversione MCCAC e altri strumenti come il compilatore COBOL Rocket Software (precedentemente Micro Focus).

Comprendi i componenti e i processi per la conversione del codice

Modernizzazione del mainframe AWS Il processo di conversione del codice include vari componenti come Modernizzazione del mainframe AWS container, bucket di progetto S3 e posizioni dei file di registro.

Argomenti

- Modernizzazione del mainframe AWS contenitore
- Bucket per progetti S3
- Posizioni dei file di registro
- Panoramica del processo

Modernizzazione del mainframe AWS contenitore

Modernizzazione del mainframe AWS Il contenitore di conversione del codice viene eseguito nel AWS CodeBuild progetto e fornisce comandi per configurare le directory del progetto e i file di configurazione, valutare il codice Assembler, espandere le macro di Assembler e convertire il codice Assembler in COBOL.

Avrai accesso al seguente AWS ECR Repository:. `381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod`

Per utilizzare le immagini, puoi seguire una delle due opzioni:

- Usa il tag più recente quando consumi l'immagine tramite AWS CodeBuild. Quando usi l'immagine, utilizzerai questo percorso:`381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod`. Ciò significa che AWS CodeBuild raccoglierà l'ultima immagine inserita nel repository.
- Elencando la versione e selezionando da quella. Per fare ciò, usa il seguente comando tramite CLI per elencare le diverse versioni nel repository:

```
aws ecr describe-images \
  --registry-id 381492161314 \
  --repository-name aws-mlogica-codebuild-prod \
  --query 'imageDetails[*].{ImagePushedAt: imagePushedAt, ImageTags: imageTags}' \
  --output json | jq '[.[] | {ImageURI: (.ImageTags[] |
 "381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod:" + .),
 ImagePushedAt: .ImagePushedAt}] | sort_by(.ImagePushedAt) | reverse'
```

  Questo elencherà tutte le immagini con il tag associato su ciascuna immagine e l'ora in cui una particolare immagine è stata rilasciata nel repository. In base al codice precedente, verrà visualizzato un elenco di immagini in cui il tag sull'immagine rappresenta la versione dell'utilità di conversione del codice. È possibile selezionare l'immagine appropriata in base alle proprie esigenze.

Bucket per progetti S3

Il codice di input e output, il codice aggiornato con macro espanse e i report generati da Modernizzazione del mainframe AWS Code conversion vengono archiviati nel bucket di progetto

che crei nel tuo. Gestione dell'account AWS Fornisci a Modernizzazione del mainframe AWS Code conversion l'accesso al bucket concedendo le autorizzazioni a un ruolo di servizio. AWS

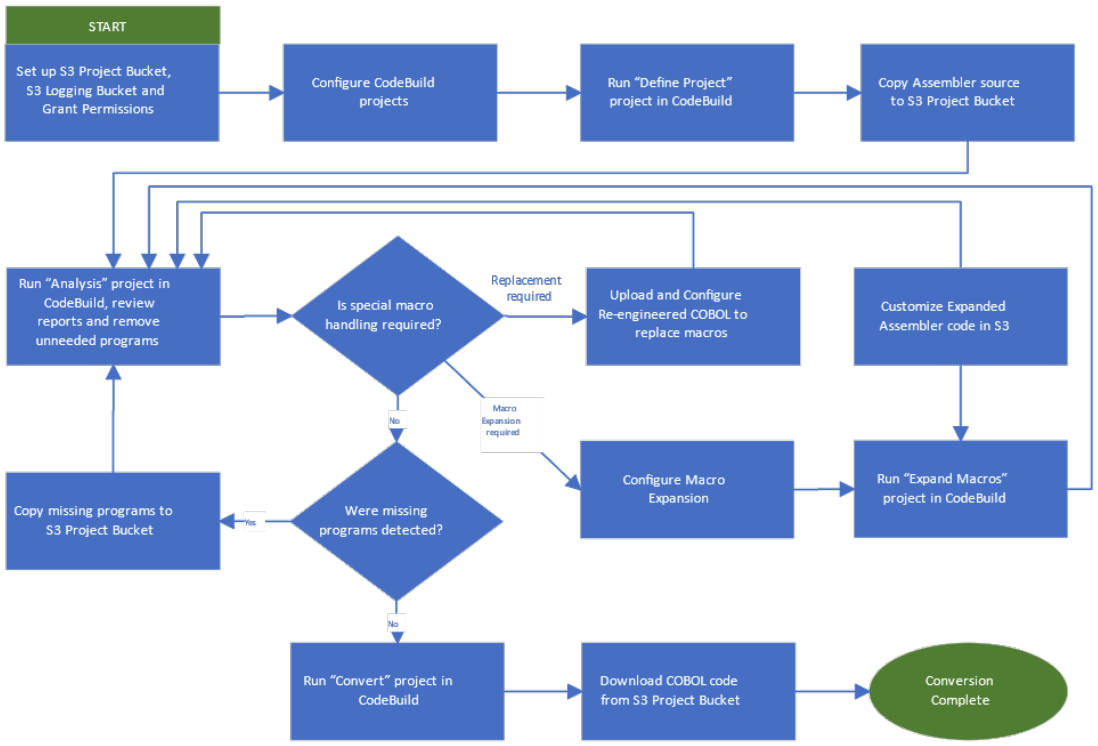
Posizioni dei file di registro

I file di registro vengono scritti in due posizioni durante l'esecuzione di ogni CodeBuild progetto:

- I file di registro con i risultati di alto livello di ogni CodeBuild passaggio vengono scritti nei file di registro nel bucket di registrazione configurato in. CodeBuild Questi file vengono visualizzati come archivi gzip con un nome di file di tipo GUID generato dal CodeBuild framework (ad es.). `0c03e183-ab40-4fe0-ba77- bc1d87e73b14.gz` Ogni archivio contiene il log generato dall'esecuzione di un CodeBuild progetto. Se l'esecuzione di un CodeBuild progetto fallisce, questo file di registro conterrà importanti informazioni sulla risoluzione dei problemi.
- I file di registro con risultati di esecuzione dettagliati a livello di componente vengono scritti nei file di registro nel percorso principale del bucket del progetto secondo lo schema del nome del file `<Project_Bucket_name>_.log` (ad esempio). `project- bucket_202406131200.log` Questi log forniscono:
  - Un riepilogo della configurazione che riporta le posizioni di input e output.
  - Un registro di ogni componente Assembler o Macro elaborato con il nome del file di destinazione.
  - Un elenco di report generati con le posizioni dei file.
  - Per le esecuzioni di conversione, un elenco dei quaderni in fase di esecuzione forniti.

Panoramica del processo

Il diagramma seguente illustra il processo di conversione di Assembler in COBOL:



Tutorial: converti il codice da Assembler a COBOL in Modernizzazione del mainframe AWS

È possibile utilizzare questo documento come step-by-step guida per comprendere come convertire il codice Assembler per la modernizzazione del mainframe in COBOL. Inoltre, puoi fare riferimento al workshop sulla conversione automatica del codice da Assembler a COBOL per saperne di più sul processo di conversione.

Argomenti

- Prerequisiti
- Passaggio 1: condividi le risorse di compilazione con Account AWS
- Fase 2: creazione di bucket Amazon S3
- Fase 3: Creare una policy IAM
- Fase 4: Creare un ruolo IAM
- Fase 5: collegare la policy IAM al ruolo IAM
- Fase 6: Creare il progetto CodeBuild
  - Fase 6.1: Creare il progetto Define

  - Fase 6.2: Creare il progetto Code Analysis
  - Fase 6.3: Creare il progetto Code Conversion
- Fase 7: Definizione del progetto e caricamento del codice sorgente
- Fase 8: Esegui l'analisi e comprendi i report
- Passaggio 9: Esegui la conversione del codice
- Passaggio 10: verifica la conversione del codice
- Passaggio 11: scarica il codice convertito
- Eseguire la pulizia delle risorse

Prerequisiti

Leggi la Comprendi la fatturazione della conversione del codice per la conversione in Assembler sezione per capire come la conversione del codice Assembler genera addebiti (report di fatturazione) e come funziona Gestione dell'account AWS la fatturazione.

Passaggio 1: condividi le risorse di compilazione con Account AWS

In questo passaggio, assicurati di condividere le risorse di compilazione con i tuoi Account AWS, specialmente nella regione in cui vengono utilizzate le risorse.

1. Apri la Modernizzazione del mainframe AWS console all'indirizzo https://console.aws.amazon.com/m2/.
2. Nella barra di navigazione a sinistra, scegli Strumenti.
3. In AWS Mainframe Modernization Code Conversion with mLogica, scegli Condividi risorse con my. Account AWS

> **Important**
>
> Devi eseguire questo passaggio una volta in ogni AWS regione in cui intendi creare build.

Fase 2: creazione di bucket Amazon S3

In questo passaggio, crei bucket Amazon S3. Il primo bucket è il bucket di progetto AWS CodeBuild per contenere il codice sorgente, quindi premi il bucket di output per contenere l' AWS CodeBuild

output (codice convertito). Per ulteriori informazioni, consulta Creazione, configurazione e utilizzo dei bucket Amazon S3 nella Amazon S3 User Guide.

1. Per creare il bucket di progetto, 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-bucket`*, dove:
   - `regionId`è la Regione AWS parte del secchio.
   - `accountId`è il tuo Account AWS ID.

   > **Note**
   >
   > Se stai creando il bucket in un paese diverso Regione AWS dagli Stati Uniti orientali (Virginia settentrionale), specifica il `LocationConstraint` parametro. 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.

Qualunque sia il nome che scegli per questi bucket, assicurati di usarli durante questo tutorial.

Fase 3: Creare una policy IAM

In questo passaggio, crei una policy IAM. La policy IAM fornita concede autorizzazioni specifiche AWS CodeBuild per l'interazione con Amazon S3, Amazon Elastic Container Registry, CloudWatch i log di Amazon che CodeBuild generano e le risorse per la conversione del codice. Amazon Elastic Compute Cloud Questa politica non è personalizzata per i clienti. La politica concede le autorizzazioni per Modernizzazione del mainframe AWS interagire e recuperare le statistiche di conversione del codice per fatturare il cliente in modo appropriato.

Per ulteriori informazioni sulla creazione di una policy IAM, consulta Creazione di politiche IAM nella guida per l'utente IAM.

Per creare una policy

1. Accedi alla console IAM e scegli Policies nel riquadro di navigazione a sinistra.
2. Scegli Crea policy.
3. Copia e incolla la seguente policy JSON nell'editor delle policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObjectAcl",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::codebuild-regionId-accountId-bucket",
                "arn:aws:s3:::codebuild-regionId-accountId-bucket/*",
                "arn:aws:s3:::aws-m2-repo-*" ],
                    "Effect": "Allow"
          },
          {
              "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "logs:*",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeVpcs",
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "*",
            "Effect": "Allow"
          }
    ]
}
```

4. Facoltativamente, puoi aggiungere tag alla politica. I tag sono coppie chiave-valore che possono aiutarti a organizzare, tracciare o controllare l'accesso alla politica.
5. Scegliere Next:Review (Successivo:Rivedi).
6. Fornisci un nome per la politica, ad esempio,. *`CodeBuildAWSM2CCMPolicy`*
7. Facoltativamente, puoi inserire una descrizione della politica e rivedere il riepilogo della politica per assicurarti che sia corretto.
8. Scegli Crea policy.

Fase 4: Creare un ruolo IAM

In questo passaggio, crei un nuovo ruolo IAM che permetta di CodeBuild 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 guida per l'utente IAM.

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, `IAMRoleTaskExecutionRoleForCodeBuild` ad esempio.
8. In Seleziona entità attendibili, verifica che il documento di policy abbia il seguente aspetto:

   JSON

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

```
            "Effect": "Allow",
            "Principal": {
              "Service": "codebuild.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
}
```

9. Scegli Crea ruolo.

Fase 5: collegare la policy IAM al ruolo IAM

In questo passaggio, colleghi la policy IAM creata in precedenza al ruolo `IAMRoleTaskExecutionRoleForCodeBuild` IAM.

1. Accedi alla console IAM e scegli Ruoli nel riquadro di navigazione a sinistra.
2. In Ruoli, scegli il ruolo che hai creato in precedenza, ad esempio`IAMRoleTaskExecutionRoleForCodeBuild`.
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. *`CodeBuildAWSM2CCMPolicy`*
5. Scegli Collega policy.

Fase 6: Creare il progetto CodeBuild

In questo passaggio, crei tre diversi CodeBuild progetti in base al `buildspec.yml` file sopra menzionato.

Fase 6.1: Creare il progetto Define

Per creare il progetto Define

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`1-awsm2ccm-define-project`.
3. Nella sezione Source, per Source provider, lascia la selezione predefinita.

4. Nella sezione Ambiente, scegli Immagine personalizzata.
5. Nel campo Tipo di ambiente, scegli Linux.
6. In Registro delle immagini, scegli Altro registro.
7. Nel campo URL del registro esterno, segui la the section called "Modernizzazione del mainframe AWS contenitore" sezione.
8. In Ruolo di servizio, scegli Ruolo di servizio esistente e, nel campo Role ARN, scegli il ruolo di servizio creato in precedenza (ad esempio,`IAMRoleTaskExecutionRoleForCodeBuild`).
9. Espandi la sezione Configurazione aggiuntiva, procedi come segue:

    a. VPC: configura se necessario in base alla configurazione.

    b. Timeout: impostato su 60 minuti.

    c. Timeout in coda: impostato su 480 minuti.

    d. Crittografia: scegli le impostazioni di crittografia appropriate (l'impostazione predefinita va bene).

    e. Nella sezione Variabili d'ambiente, aggiungi quanto segue uno per uno:

    - Nome: PROJECT_BUCKET. Valore:. **`codebuild-regionId-accountId- bucket`** Tipo: testo semplice
    - Nome: PROJECT_DIR. Valore:. **`prj_codebuild_01`** Tipo: testo semplice
    - Nome: AWSM2 CCM_ACTION. Valore:. **`define_project`** Tipo: testo semplice
    - Nome: AWSM2 CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Valore:. Tipo: testo semplice

10. Nella sezione Buildspec, scegliete Inserisci comandi di compilazione, quindi Passa all'editor.
11. Sostituisci i valori correnti con questo:

```
version: 0.2
phases:
    build:
        commands:
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '**/*'
    discard-paths: no
    base-directory: $PROJECT_DIR
```

dove, PROJECT_DIR sono variabili di ambiente disponibili all'interno. CodeBuild Per ulteriori informazioni, consulta Variabili di ambiente negli ambienti di compilazione.

12. Nella sezione Artefatti, procedi nel seguente modo:

    - in Tipo, scegli Amazon S3, quindi scegli il tuo bucket di output, ad esempio. `codebuild-regionId-accountId-bucket`
    - per Path, lascia vuoto questo campo.
    - per Nome, immettere**`prj_codebuild_01`**.
    - per la confezione di Artifact, seleziona Nessuno.
    - per Sostituisci il nome dell'artefatto, deseleziona questa opzione.
    - per la crittografia, lasciala alle impostazioni predefinite.

13. Per la sezione Registri, procedi come segue:

    - CloudWatch registri: disabilitati
    - Registri S3: abilitati

    - Secchio: **`codebuild-regionId-account-bucket`**
    - Percorso del registro: **`CODEBUILD-LOGS`**

14. Scegliere Create build project (Crea progetto di compilazione).

Fase 6.2: Creare il progetto Code Analysis

Per creare il progetto Code Analysis

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`2-awsm2ccm-analysis`.
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-bucket`
4. Nel campo della chiave oggetto S3 o della cartella S3, inserisci. **`prj_codebuild_01`**
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, segui la the section called "Modernizzazione del mainframe AWS contenitore" sezione.
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,`IAMRoleTaskExecutionRoleForCodeBuild`).
10. Espandi la sezione Configurazione aggiuntiva, procedi come segue:
    a. VPC: configura se necessario in base alla configurazione.
    b. Timeout: impostato su 60 minuti.
    c. Timeout in coda: impostato su 480 minuti.
    d. Crittografia: scegli le impostazioni di crittografia appropriate (l'impostazione predefinita va bene).
    e. Nella sezione Variabili d'ambiente, aggiungi quanto segue uno per uno:
        - Nome: PROJECT_BUCKET. Valore:. **`codebuild-regionId-accountId-bucket`** Tipo: testo semplice
        - Nome: PROJECT_DIR. Valore:. **`prj_codebuild_01`** Tipo: testo semplice
        - Nome: AWSM2 CCM_ACTION. Valore:. **`analysis`** Tipo: testo semplice
        - Nome: AWSM2 CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Valore:. Tipo: testo semplice
11. Nella sezione Buildspec, scegli Inserisci comandi di compilazione, quindi Passa all'editor.
12. Sostituisci i valori correnti con questo:

```
version: 0.2
phases:
    build:
        commands:
            - ln -s $CODEBUILD_SRC_DIR $PROJECT_DIR
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '*.log'
        - '_Converted/*/*'
        - '_Reports/*'
    secondary-artifacts:
        reports:
            files:
                - '_Reports/AWSM2CCM*'
```

```
    discard-paths: no
    base-directory: $PROJECT_DIR
```

dove, PROJECT_DIR sono variabili di ambiente disponibili all'interno. CodeBuild Per ulteriori informazioni, consulta Variabili di ambiente negli ambienti di compilazione.

13. Nella sezione Artefatti, procedi nel seguente modo:
    - in Tipo, scegli Amazon S3, quindi scegli il tuo bucket di output (ad es.). `codebuild-regionId-accountId-bucket`
    - per Path, inserisci ARTIFACTS.
    - per Nome, immettere. **`prj_codebuild_01`**
    - per la confezione di Artifact, seleziona Nessuno.
    - per Sostituisci il nome dell'artefatto, deseleziona questa opzione.
    - per la crittografia, lasciala alle impostazioni predefinite.
14. Per la sezione Registri, procedi come segue:
    - CloudWatch registri: disabilitati
    - Registri S3: abilitati
    - Secchio: **`codebuild-regionId-account-bucket`**
    - Percorso del registro: **`CODEBUILD-LOGS`**
15. Scegliere Create build project (Crea progetto di compilazione).

Fase 6.3: Creare il progetto Code Conversion

Per creare il progetto Code Conversion

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,`3-awsm2ccm-convert`).
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-bucket`
4. Nel campo della chiave oggetto S3 o della cartella S3, inserisci. **`prj_codebuild_01`**
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, segui la the section called "Modernizzazione del mainframe AWS contenitore" sezione.
9. In Ruolo di servizio, selezionare Ruolo di servizio esistente e, nel campo Role ARN, scegliere il ruolo di servizio creato in precedenza, ad esempio. `IAMRoleTaskExecutionRoleForCodeBuild`
10. Espandi la sezione Configurazione aggiuntiva, procedi come segue:
    a. VPC: configura se necessario in base alla configurazione.
    b. Timeout: impostato su 60 minuti.
    c. Timeout in coda: impostato su 480 minuti.
    d. Crittografia: scegli le impostazioni di crittografia appropriate (l'impostazione predefinita va bene).
    e. Nella sezione Variabili d'ambiente, aggiungi quanto segue uno per uno:
       - Nome: PROJECT_BUCKET. Valore:. **`codebuild-regionId-accountId-bucket`** Tipo: testo semplice
       - Nome: PROJECT_DIR. Valore:. **`prj_codebuild_01`** Tipo: testo semplice
       - Nome: AWSM2 CCM_ACTION. Valore:. **`conversion`** Tipo: testo semplice
       - Nome: AWSM2 CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Valore:. Tipo: testo semplice
11. Nella sezione Buildspec, scegli Inserisci comandi di compilazione, quindi Passa all'editor.
12. Sostituisci i valori correnti con questo:

```
version: 0.2
phases:
    build:
        commands:
            - export AWSM2CCM_PUSH_RUNTIME_COPYBOOKS=y
            - ln -s $CODEBUILD_SRC_DIR $PROJECT_DIR
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '*.log'
```

```
        - '_Converted/*/*'
        - '_Reports/*'
    discard-paths: no
    base-directory: $PROJECT_DIR
```

dove, PROJECT_DIR sono variabili di ambiente disponibili all'interno. CodeBuild Per ulteriori informazioni, consulta Variabili di ambiente negli ambienti di compilazione.

13. Nella sezione Artefatti, procedi nel seguente modo:

    - in Tipo, scegli Amazon S3, quindi scegli il tuo bucket di output (ad es.). `codebuild-regionId-accountId-bucket`
    - per Path, inserisci ARTIFACTS.
    - per Nome, immettere. **`prj_codebuild_01`**
    - per la confezione di Artifact, seleziona Nessuno.
    - per Sostituisci il nome dell'artefatto, deseleziona questa opzione.
    - per la crittografia, lasciala alle impostazioni predefinite.
14. Per la sezione Registri, procedi come segue:

    - CloudWatch registri: disabilitati
    - Registri S3: abilitati

    - Secchio: **`codebuild-regionId-account-bucket`**
    - Percorso del registro: **`CODEBUILD-LOGS`**
15. Scegliere Create build project (Crea progetto di compilazione).

Fase 7: Definizione del progetto e caricamento del codice sorgente

Il Define Project imposta la cartella del progetto e i file di configurazione, inizializzati con configurazioni predefinite. In questo passaggio, si avvia la compilazione. Per farlo:

1. Accedi alla AWS CodeBuild console.
2. Nel riquadro di navigazione a sinistra scegli Crea progetti.
3. Seleziona il progetto creato in precedenza (`1-awsm2ccm-define-project`) da creare
4. Scegliete Avvia compilazione, quindi Avvia ora per definire il progetto. Una volta avviata la compilazione, lo stato passerà a In corso.

5. Scegli i dettagli della fase per vedere lo stato di avanzamento di ogni fase orchestrata dal AWS CodeBuild progetto.
6. Attendi che lo stato sia cambiato in riuscito per tutti i passaggi.
7. Vai alla console Amazon S3.
8. Individua e fai clic sul bucket Amazon S3 denominato `codebuild-regionId-accountId-bucket`

   - **`CODEBUILD-LOGS/`**la cartella contiene i AWS CodeBuild log dei progetti in esecuzione. AWS CodeBuild
   - **`prj_codebuild_01/`**cartella che contiene la struttura del progetto. Viene utilizzata durante le fasi di analisi, expand_macros e conversione. Puoi scegliere di esplorare i dettagli `prj_codebuild_01/`
   - **`cobol_reserved.rsw`**file di configurazione (elenco di parole COBOL) riservato al convertitore. Viene utilizzato durante la fase di conversione.
   - **`Macro_Expansion/`**la cartella contiene macro da espandere nei programmi Assembler. Viene utilizzata durante la fase expand_macros.
   - **`macro_settings.json`**il file di configurazione contiene macro sostitutive personalizzate. Viene utilizzato durante la fase expand_macros.
   - **`macrolib/`**la cartella contiene le macro di Assembler da convertire. Viene utilizzata durante la fase di analisi e conversione.

     1. Seleziona `macrolib/`.
     2. Per impostazione predefinita, `MACRO1.mac` viene fornita come file di esempio una macro Assembler denominata. Eliminate questo file poiché non è necessario per l'analisi.
     3. Carica le tue macro in questa directory.
   - **`project_settings_aux.json`**il file di configurazione contiene le impostazioni relative alla tabella codici. Viene utilizzato durante la fase di conversione.
   - **`project_settings.json`**il file di configurazione contiene le impostazioni per il convertitore. Viene utilizzato durante la fase di conversione.
   - **`srclib/`**la cartella contiene i programmi Assembler da convertire. Viene utilizzata durante la fase di analisi e conversione.

     1. Scegli `srclib/`.

2. Per impostazione predefinita, due programmi Assembler `SQtest02.asm` denominati `SQtest01.asm` e vengono forniti come esempi. Eliminate questi file perché non sono necessari per l'analisi e la conversione.
3. Carica i tuoi programmi Assembler in questa directory.

9. Verifica lo stato del `1-awsm2ccm-define-project` passaggio. Dovrebbe aver avuto successo nella scheda Stato dell'ultima build.

Sei pronto per il passaggio successivo: analisi del codice.

Fase 8: Esegui l'analisi e comprendi i report

> **Note**
>
> Modernizzazione del mainframe AWS La fase di analisi della conversione del codice è gratuita.

In questo passaggio, dai il via a un'altra build:

1. Nel riquadro di navigazione a sinistra, scegli Crea progetti.
2. Scegli il progetto che hai creato nel passaggio 6.2 per creare:`2-awsm2ccm-analysis`.
3. Scegli Avvia compilazione, quindi Inizia ora per generare report di analisi. Questo avvierà la compilazione e cambierà lo stato in corso.
4. Scegli i dettagli della fase in cui vedrai lo stato di avanzamento di ogni fase orchestrata dal AWS CodeBuild progetto. Attendi che lo stato cambi a completato per tutti i passaggi.
5. Da Console di gestione AWS, vai alla console di servizio Amazon S3.
6. Individua e fai clic sul bucket Amazon S3: `codebuild-regionId-accountId-bucket`

    a. **`ARTIFACTS/`**la cartella contiene gli output delle fasi di analisi e conversione.

    b. Scegli `ARTIFACTS/prj_codebuild_01/_Reports/`.

    c. Saranno disponibili i seguenti report:

    - `AWSM2CCM-Analysis-Report-<timestamp>.pdf`è un rapporto esecutivo che illustra la fatturazione e l'ambito della conversione del Modernizzazione del mainframe AWS Codice, il miglioramento della conversione, il riepilogo delle conversioni e le statistiche

dettagliate sulle conversioni. Riepiloga inoltre i conteggi dei codici e i conteggi dei codici fatturabili a livello di progetto e fornisce metriche ed elenchi dei membri di riferimento per ciascun componente. È fondamentale eseguire ed esaminare questo rapporto prima di eseguire la conversione effettiva.

- `Conversion_Detailed_Statistics.txt`fornisce la frequenza e il risultato di conversione previsto (indicato come «Stato di conversione») per ogni istruzione presente in ogni componente. In questo modo è possibile identificare rapidamente se le istruzioni sono chiare e non supportate dal convertitore. I possibili risultati dello stato di conversione sono:
  - Completamente convertita: l'istruzione verrà convertita accuratamente in COBOL.
  - Convertita parzialmente: l'istruzione è supportata ma utilizza un parametro o un'espressione non supportati. È probabile che dopo la conversione siano necessarie regolazioni manuali.
  - Non convertita: l'istruzione non è supportata dal convertitore.
  - Istruzioni di precompilazione per la verifica: queste sono normalmente incluse nelle macro e si riferiscono a quelle che probabilmente sono note anche come istruzioni del Conditional Assembly Language (ad esempio, AIF, AGO) sul mainframe. Queste sono gestite dal precompilatore, che si basa su tali istruzioni o direttive, seleziona e produce codice ASM pulito/statico. Queste istruzioni dipendono dai valori effettivi dei parametri Macro che vengono compilati. Quindi, la stessa Macro può generare diverse parti di codice ASM, a seconda dei valori dei parametri passati. Ciò è dovuto alla presenza di tali istruzioni di precompilazione. In tal caso, valuta la possibilità di espandere o riprogettare la Macro.
- `Conversion_Global_Statistics.txt`fornisce un riepilogo dello stato di conversione a livello di componente.
- `CrossReference_PgmToCpyMacro.txt`riporta le dipendenze del programma Assembler sulle macro. Fornisce un modo rapido per determinare se mancano delle macro nel codice caricato.
- `CrossReference_PgmToPgm.txt`riporta le dipendenze del programma Assembler da altri programmi Assembler. Fornisce un modo rapido per determinare se nel codice caricato mancano programmi Assembler.

7. Torna alla console di AWS CodeBuild servizio.
8. Verifica lo stato della fase di analisi 2-awsm2ccm. Avrebbe dovuto avere successo nella scheda Stato della build più recente.

Sei pronto per il passaggio successivo: conversione del codice.

Passaggio 9: Esegui la conversione del codice

> **Important**
>
> Modernizzazione del mainframe AWS La fase di conversione del codice verrà fatturata in base all'utilizzo. Per ulteriori informazioni sulla fatturazione, consulta the section called "Comprendi la fatturazione con conversione del codice".

In questo passaggio, configurerai il processo di conversione e quindi avvierai la build.

1. Da Console di gestione AWS, vai al servizio Amazon S3.
2. Individua e fai clic sul bucket Amazon S3:. `codebuild-regionId-accountId-bucket`
   a. Passa a `prj_codebuild_01/`.
   b. Seleziona **`project_settings.json`** e scegli Scarica.
   c. Apri il `project_settings.json` file per visualizzare la seguente struttura JSON:

```
{
"Source programs directory":"srclib",
"Source copybooks/macros directory":"macrolib",
"Copybook/Macros Conversion":"Called_only",
"Do not regenerate the Copy/Macro if already exists":"false",
"Target Compiler":"IBM",
"Endianess":"Big",
"Converted programs extension":"",
"Converted CICS programs extension":"",
"Converted copies/macros extension":"",
"Trace Level":"STANDARD",
"Trace file open mode":"append",
"Data definition level":5,
"Start picture column":40,
"Generate Sync FILLER with name":"FILL-SYNC",
"Use SYNC clause":"yes",
"Decimal Point Comma":"true",
"Original Source Placement":"RIGHT"
}
```

dove:

- Directory del programma di origine: contiene i programmi Assembler necessari per la conversione.
- Directory Source Copybooks/Macros: contiene le macro e i quaderni Assembler necessari per la conversione.
- La conversione Copybooks/Macros può essere una delle seguenti:
  - Tutto: questo pulsante di opzione indica che la conversione completa convertirà tutto ciò che è copybook/Macros disponibile nella directory indipendentemente dal fatto che sia utilizzato dai programmi o meno.
  - Called_only: Questo pulsante di opzione indica che la conversione completa convertirà solo quello effettivamente utilizzato dai programmi. copybook/Macros
-

  > **Important**
  >
  > Non è necessario rigenerarlo se esiste già. Copy/Macro

  Se questo è vero, lo strumento non converte nuovamente il CopyBook/macro, se è già stato convertito (esiste nella cartella di output).
- Obiettivo: la conversione dei programmi (codice generato) dipende dal compilatore COBOL di destinazione. Sono supportate le seguenti opzioni:
  - «IBM» per mainframe IBM
  - «MF» per Micro Focus COBOL
  - «VERYANT» per Veryant ISCOBOL
  - «NTT» per NTT DATA Enterprise COBOL (Unikix)
- Endianess e Bitness: La conversione dei programmi (codice generato) dipende dalla piattaforma di destinazione (bit/endianess). Questa combinazione consente la selezione delle seguenti opzioni supportate:
  - Endianess: Big (per Big-Endian)/Little (Little-Endian). Ad esempio, il z/OS mainframe IBM è Big-Endian, Windows è Little-Endian, Linux varia in base alla distribuzione (ad esempio Amazon Linux 2 su EC2 è Little-Endian).
  - Bitness: 32/64 (se non viene fornito, il valore predefinito sarà 32). L'impostazione consigliata è 32 bit.

- Estensione del programma convertita: consente di impostare l'estensione del file per i programmi COBOL generati. Vuoto («»): nessuna estensione. Per gli obiettivi COBOL di Rocket Software (precedentemente Micro Focus), si consiglia CBL per consentire a Rocket Enterprise Developer di riconoscere correttamente i file.
- Estensione del programma CICS convertita: consente di impostare l'estensione del file per i programmi CICS COBOL generati. Vuoto («»): nessuna estensione. Per i target COBOL di Rocket Software, si consiglia CBL per consentire a Rocket Enterprise Developer di riconoscere correttamente i file.
- Estensione Copybooks/Macros convertita: consente di impostare l'estensione del file per i quaderni COBOL generati. Empty («»): nessuna estensione. Per i target COBOL di Rocket Software, si consiglia CPY per consentire a Rocket Enterprise Developer di riconoscere correttamente i file.
- Livello di traccia: Trace è l'informazione che viene registrata e utilizzata durante la conversione. CodeBuild L'utente può selezionare il livello di dettaglio selezionando una delle opzioni fornite.
  - ERROR = TRACE ERROR: vengono visualizzati solo gli errori di conversione.
  - STANDARD = TRACE STANDARD: vengono visualizzati gli errori di conversione e le informazioni standard. Questa è l'impostazione consigliata.
  - ALL = TRACE ALL: livello massimo di tracciamento
- Modalità di apertura del file di traccia: non utilizzata. Si consiglia l'impostazione predefinita di append.
- Livello di definizione dei dati: indica il livello iniziale dei sottocampi (dopo il livello «01") definito nella sezione working-storage e linkage. Deve essere un numero.
- Colonna dell'immagine iniziale: riguarda il formato del codice COBOL generato e indica la colonna in cui è inserita la clausola PIC (dopo i nomi dei campi). Deve essere un numero.
- Posizionamento della fonte originale: indica la posizione in cui vengono inseriti i commenti nel programma. Ha due opzioni:
  - DESTRA: Questa opzione posizionerà il commento o le informazioni aggiuntive nella giusta posizione dopo la settantatreesima (73) colonna. In COBOL il codice è scritto nelle prime settantadue (1-72) colonne e qualsiasi elemento compreso nella settantatreesima (>= 73) colonna verrà considerato come un commento.
  - SOPRA: Questa opzione posizionerà il commento sopra il contenuto tradotto.
- Genera Sync FILLER con nome: Questa opzione è correlata all'allineamento in memoria dei campi binari (tipi di dati Assembler «H», «F», «D», che vengono convertiti nel tipo

di dati COBOL «COMP»). Per garantire il corretto limite di allineamento, durante la conversione verranno aggiunti campi di riempimento espliciti. Questa è un'opzione basata su testo, il valore deve essere una stringa (come FILL-SYNC).

- Usa la clausola SYNC: questa opzione si riferisce all'allineamento in memoria dei campi binari. Sì = tutti i campi sono stati convertiti in COBOL. «COMP» sarà definito con la clausola «SYNC» (ad esempio, 05 WRKFLD PIC S9 (09) COMP SYNC).
- Virgola decimale: se questo è vero, la clausola DECIMAL-POINT IS COMMA verrà aggiunta al paragrafo COBOL «SPECIAL-NAMES».

d. In base alle tue esigenze, modifica i parametri appropriati, quindi salva il. `project_settings.json`

e. Rimuovi il `project_settings.json` file esistente dal `prj_codebuild_01/` bucket Amazon S3, quindi carica la nuova versione.

3. Torna al servizio. AWS CodeBuild
4. Seleziona il progetto da costruire che hai creato in precedenza: `3-awsm2ccm-convert`

a. Scegliete Avvia compilazione, quindi Avvia ora per convertire i programmi e le macro di Assembler in programmi e quaderni COBOL.

b. Attendi che lo stato di compilazione cambi in Riuscito per questo progetto. Si troverà nella scheda Stato della build più recente.

Passaggio 10: verifica la conversione del codice

1. Da Console di gestione AWS, vai al servizio Amazon S3.
2. Individua e fai clic sul bucket Amazon S3:. `codebuild-regionId-accountId-bucket`
3. Vai a. **`awsm2ccm-do-not-delete`** Modernizzazione del mainframe AWS La conversione del codice crea file binari codificati per ogni modulo Assembler o Macro durante il processo di conversione. Questi file sono essenziali per prevenire la duplicazione della fatturazione ai clienti e anche per tenere traccia della quantità di codice Assembler fornito che è stata analizzata e convertita. I file vengono archiviati nella seguente posizione:. `codebuild-regionId-accountId- bucket/awsm2ccm-do-not-delete/<your_AWS_account_id>/Hash` I file codificati non contengono alcun codice Assembler e inoltre non è possibile estrarre il codice cliente da questi file.

> **Important**
>
> Né modificare manualmente questi file né eliminarli. La modifica o l'eliminazione di questi file può comportare fatturazioni multiple per gli stessi componenti.

Tratta la **awsm2ccm-do-not-delete/**cartella come una directory gestita dal sistema. Consultate Supporto prima di apportare modifiche a questa directory o al suo contenuto.

4. Fai clic `codebuild-regionId-accountId-bucket` per tornare al bucket.
5. Scegli **ARTIFACTS/prj_codebuild_01/**. La cartella _Converted/ contiene gli output COBOL generati come risultato della fase di conversione del codice. Avrà le seguenti sottodirectory:
   - la cartella copybooks/ contiene i quaderni COBOL generati.
   - la cartella programs/ contiene i programmi COBOL generati.
   - La cartella runtime_lib/ contiene programmi e quaderni COBOL aggiuntivi forniti dalla soluzione.
6. Se i rapporti di analisi e gli altri rapporti indicano che la conversione è avvenuta correttamente e il AWS CodeBuild progetto **3-awsm2ccm-convert** è contrassegnato come completato, scaricate il codice COBOL e i quaderni dalla directory _Converted/.

Passaggio 11: scarica il codice convertito

In questo passaggio, scaricate il codice COBOL e i quaderni dalla directory _Converted/ e compilateli nell'ambiente COBOL di destinazione.

1. Da Console di gestione AWS, vai al servizio Amazon S3.
2. Individua e fai clic sul bucket Amazon S3:. `codebuild-regionId-accountId-bucket`
3. Vai alla posizione:. `ARTIFACTS/prj_codebuild_01/_Converted/`
4. Scarica il codice COBOL convertito da tutte le sottodirectory in _Converted/. Puoi anche usare il seguente comando CLI per scaricarli contemporaneamente:

```
aws s3 cp s3://codebuild-regionId-accountId-
bucket/ARTIFACTS/prj_codebuild_01/_Converted/ . --recursive
```

5. Analizza e compila il COBOL convertito nell'ambiente COBOL di destinazione.

Eseguire la pulizia delle risorse

Se non hai più bisogno delle 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 politiche IAM nella guida per l'utente IAM.
- Elimina il ruolo IAM che hai creato per questo tutorial. Per ulteriori informazioni, consulta Eliminazione di ruoli o profili di istanza nella guida per l'utente 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.

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 i modelli 5, 10 e 20. SPARCstation 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 necessità 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 dal Sun SPARCstation 20): originariamente, una variante multiprocessore Sun-4, basata sul bus del modulo processore introdotto nella serie 600MP. MBus SPARCServer L'architettura Sun-4m in seguito comprendeva anche sistemi non MBus uniprocessori come SPAR Cstation 5, che utilizzavano processori con architettura SPARC V8. Supportato a partire da SunOS 4.1.2 e da Solaris 2.1 a Solaris 9. SPARCServer Il supporto a 600 MP è stato interrotto dopo Solaris 2.5.1.

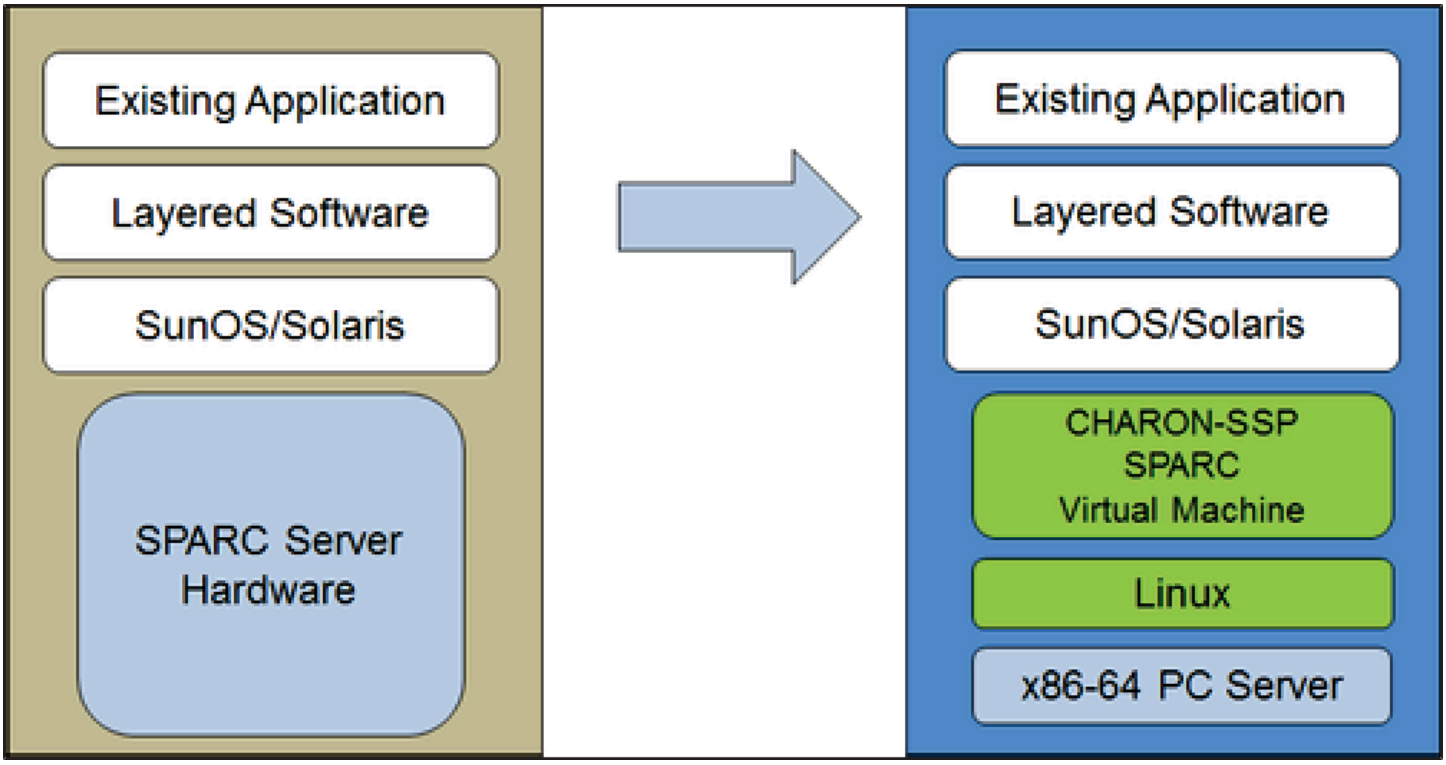
Charon-SSP/4U (+) emula il seguente hardware SPARC:

- 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 UltraSP ARC 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 il supporto AMD-di Intel e AMD e devono essere installate su un sistema host dedicato. VTx/EPT v/NPT hardware

assisted virtualization technology in modern CPUs to offer better virtual CPU performance. Charon-SSP/4U+ and Charon-SSP/4V+ require CPUs with VT-x/EPT or AMD-v/NPT L'esecuzione di queste varianti di prodotto in una macchina virtuale (ad esempio, on VMware) non è supportata.

> **Note**
>
> Se prevedi di eseguire Charon-SSP/4U+ o 4V+ in un ambiente cloud, contatta Stromasys o un Stromasys VAR 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 I/O deve essere aumentato se le applicazioni guest producono un carico elevato I/O . Inoltre, un sistema con molti emulati CPUs è in genere in grado di creare un I/O carico maggiore e quindi I/O potrebbe essere necessario aumentare il numero di core CPU disponibili. In un ambiente hyper-threading, per ottenere prestazioni ottimali, il numero di core CPU (ovvero real/physical CPUs) 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 i core emulati CPUs e i core CPU per l'elaborazione è determinata dalla configurazione. I/O 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 istanza size/type 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' I/O elaborazione (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 all'elaborazione 1/3 (min. 1; arrotondato per difetto) del numero di elementi CPUs visibili all'host Charon. 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. In più:
  - 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 sul moderno x86-64 CPUs, due thread possono essere eseguiti su un core fisico della CPU fornendo due elementi logici al sistema operativo host. CPUs Se possibile, disabilita l'hyper-threading sull'host Charon-SSP. Tuttavia, spesso ciò non è possibile negli VMware ambienti 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- (istanze baremetal) e pertanto non possono essere eseguiti in tutti gli ambienti cloud. x/ EPT or AMD-v/NPT 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 Console di gestione AWS . 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 Console di gestione AWS 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 Console di gestione AWS per avviare una nuova istanza

Per creare una nuova istanza

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. 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 altro AMIs. 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 il tipo di chiave SSH 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 scegli Modifica. Scegliete le impostazioni che corrispondono al vostro 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 dispone di 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.

8. 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.

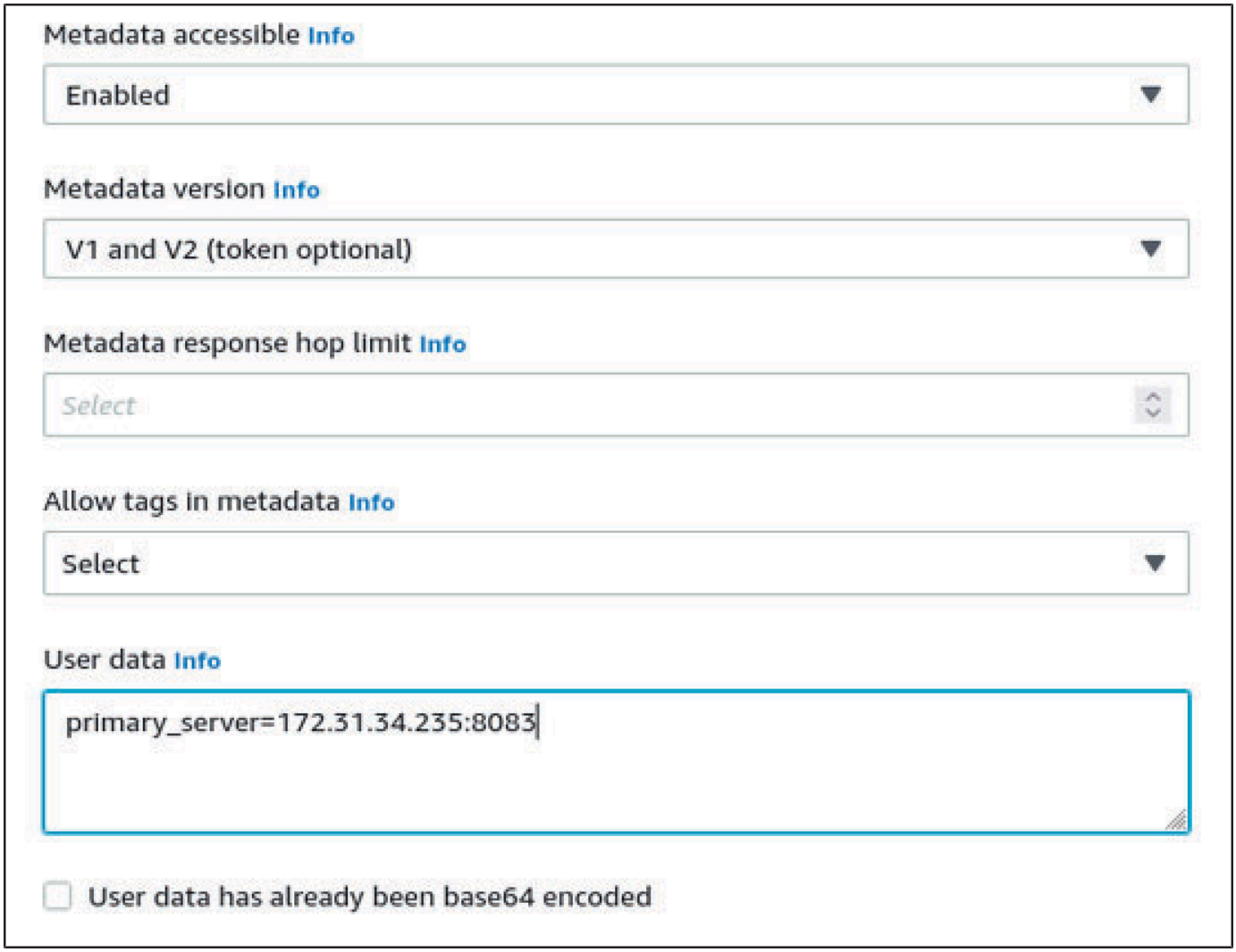
9. Espandi la sezione Dettagli avanzati, scorri verso il basso e seleziona Specificare le opzioni della CPU. Tre esempi che hanno maggiori probabilità di essere utili in un ambiente di emulazione Charon sono mostrati nell'immagine seguente.



10. 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.



    Le seguenti sono opzioni di configurazione dei dati utente valide:

    - **primary_server=**<*ip-address*>[**:**<*port*>]
    - **backup_server=**<*ip-address*>[**:**<*port*>]

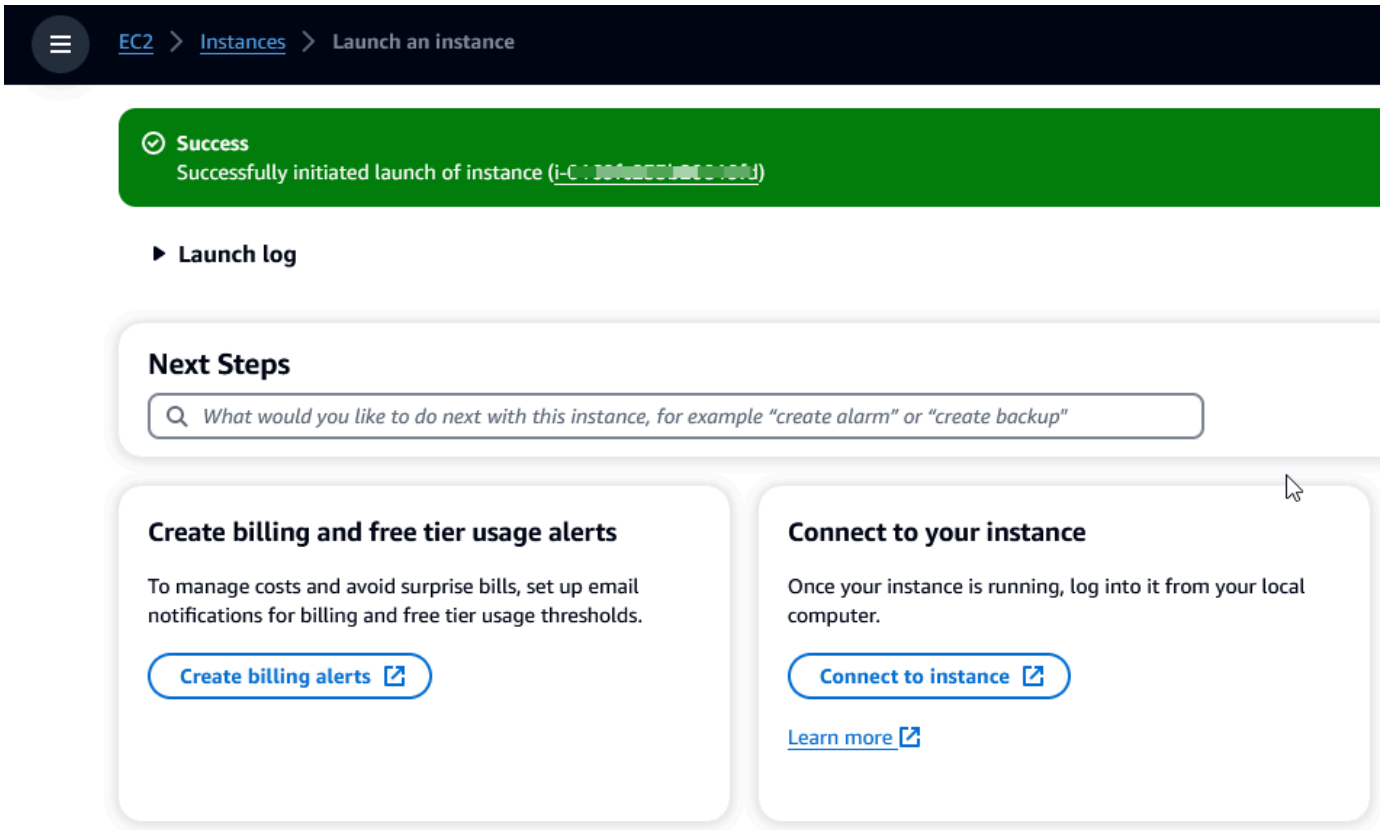
    Dove

    - <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:



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 Amazon EC2 User Guide.

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 accedervi e utilizzarle. AMIs

Prerequisiti

- Assicurati di avere accesso da amministratore a un AWS account in cui 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.

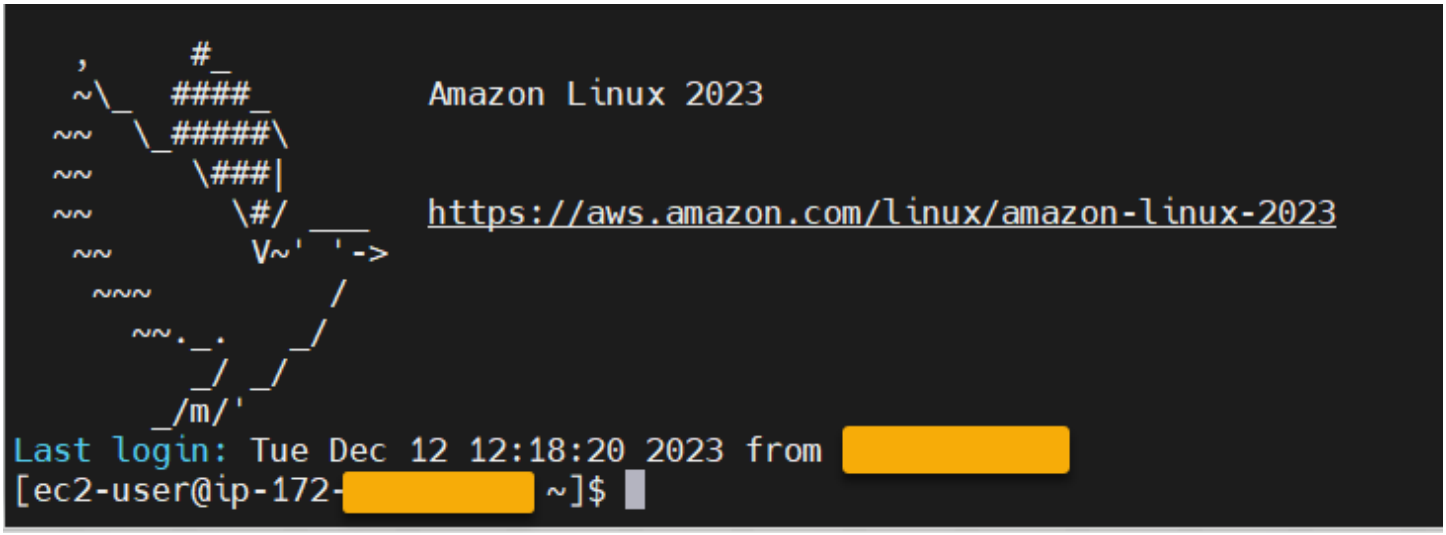
1. Accedi Console di gestione AWS e apri la Marketplace AWS console su https://console.aws.amazon.com/marketplace.
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).
3. Copia e incolla il seguente link nella barra degli indirizzi del browser: https://aws.amazon.com/marketplace/ pp/prodview-eg227ymldsnx2
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

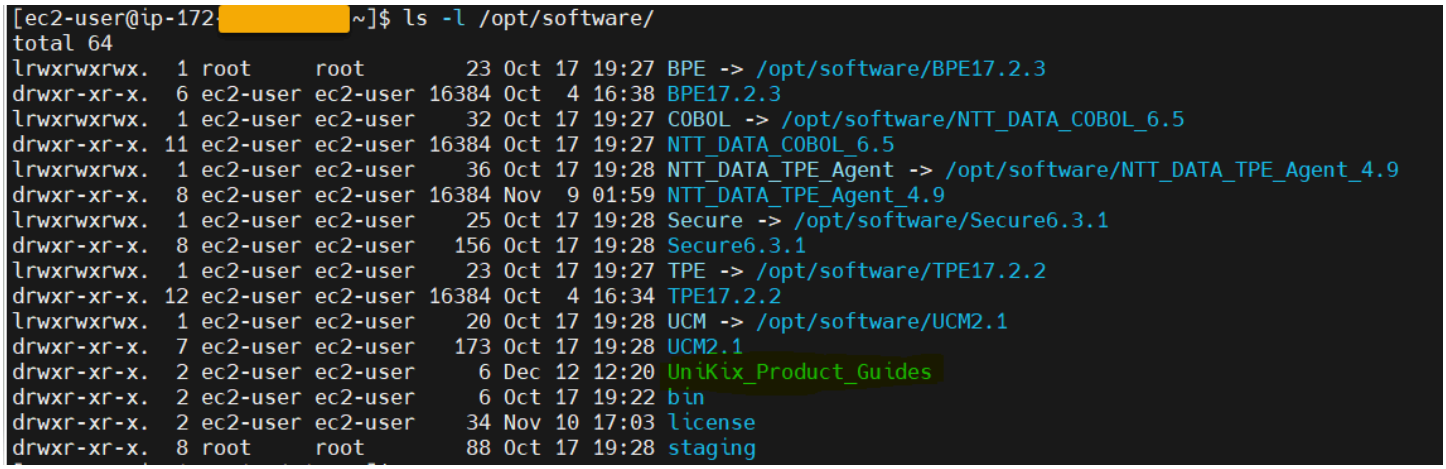
1. Apri la console su /marketplace Marketplace AWS . https://console.aws.amazon.com
2. Nel riquadro di navigazione a sinistra, scegli Gestisci abbonamenti.
3. Trova l'AMI che desideri avviare e scegli Launch new instance.
4. In Regione, seleziona la regione consentita.
5. Scegli Continua a lanciare tramite 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 Guida per l'utente di Amazon EC2.
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 di Rocket Software (precedentemente 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.



Sotto la `/opt/software/` cartella, c'è una cartella denominata`UniKix_Product_Guides`, come mostrato nell'immagine seguente.



La `UniKix_Product_Guides` cartella include la documentazione per i seguenti componenti installati su questa istanza Amazon EC2:

- TIPO DI DATI NTT
- DATI NTT BPE
- NTT DATA Enterprise COBOL
- NTT DATA Secure UniKix
- Gestore centrale NTT DATA UniKix

La `software` cartella che appare nell'immagine precedente contiene i file binari per i componenti elencati sopra.

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.

Tutorial: distribuisci l'applicazione su NTT DATA CardDemo

Questa pagina illustra il step-by-step processo di distribuzione dell'applicazione di CardDemo esempio sulla ripiattaforma AWS Mainframe Modernization con il runtime NTT DATA Unikix.

L'applicazione CardDemo di esempio è un'applicazione mainframe semplificata progettata e sviluppata per testare e presentare la tecnologia dei partner per casi AWS d'uso di migrazione e modernizzazione del mainframe.

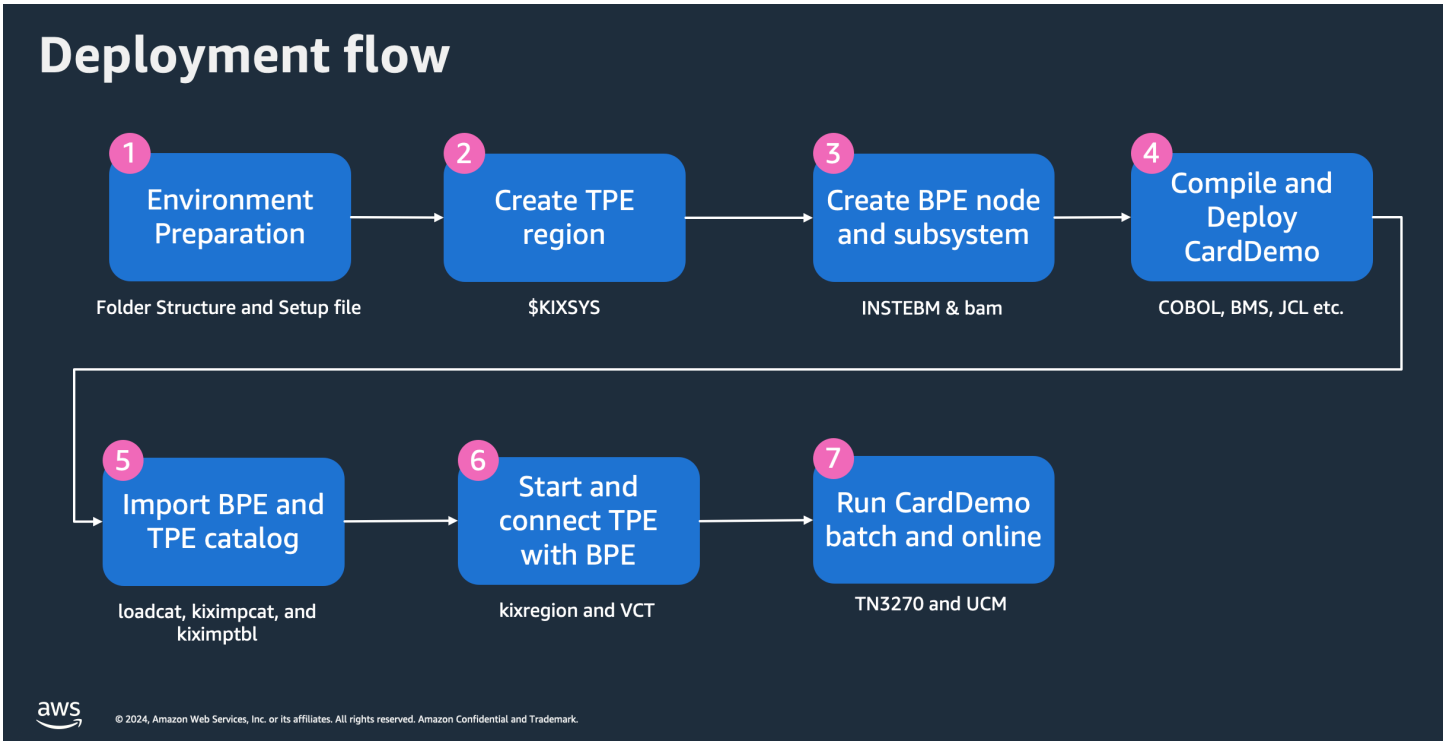
Per ulteriori informazioni su questa applicazione, vedere, repository for. GitHub CardDemo

Argomenti

- Diagramma di flusso di distribuzione
- Prerequisiti
- Fase 1: Preparare l'ambiente
- Fase 2: Creare una regione TPE
- Fase 3: Creare il nodo e il sottosistema BPE
- Fase 4: Compilare e distribuire l'applicazione CardDemo
- Fase 5: Importazione del catalogo BPE e TPE
- Fase 6: Avviare e collegare TPE con BPE
- Fase 7: Eseguire l'applicazione CardDemo
- Risoluzione dei problemi

Diagramma di flusso di distribuzione

Il diagramma seguente mostra ogni fase del flusso di lavoro per la distribuzione di un'applicazione sul runtime NTT DATA Unikix.



Prerequisiti

- Segui le istruzioni fornite nell'Ripiattaforma con NTT DATAutilizzo dell'AMI NTT DATA UniKix Marketplace.
- Modifica l'opzione dei metadati dell'istanza IMDSv2su Opzionale, come indicato in Ripristina l'uso di IMDSv1 nella guida per l'utente di Amazon EC2.
- Scarica i componenti CardDemo di runtime per NTT DATA dal UniKix

  GitHub archivio.
- Accedi all'istanza EC2 in UniKix fase di esecuzione come. `ec2-user`
- Estrai i componenti CardDemo di runtime scaricati utilizzando questo link: UniKix_ CardDemo _runtime_v1.zip.
  - La directory estratta deve contenere `bin` e `migrated_app` directory.
  - Sposta entrambe `bin` le `migrated_app` cartelle nella tua cartella. `$HOME` Il percorso sarebbe simile `/home/ec2-user` a.
  - Dovresti avere le seguenti directory nella tua`$HOME`:
    - `/home/ec2-user/bin`
    - `/home/ec2-user/migrated_app`

- Sposta tutti i file all'interno della directory $HOME/bin con il seguente comando:
- • `chmod +x $HOME/bin/*`

Fase 1: Preparare l'ambiente

Dopo aver completato i prerequisiti, il primo passaggio consiste nel preparare l'ambiente in cui si desidera distribuire l' CardDemo applicazione.

1. Accedi all'istanza EC2 in UniKix fase di esecuzione come. `ec2-user`
2. Osserva l'elenco dei UniKix software preconfezionati all'interno dell'AMI, come TPE, BPE e COBOL, insieme ad altri presenti nella posizione del UniKix prodotto NTT DATA utilizzando il seguente comando nell'istanza EC2:

   ```
   ls -l /opt/software/
   ```

3. Esamina l'applicazione migrata. CardDemo Vedrai tutto il codice sorgente, comprese le mappe BMS, i programmi COBOL, i quaderni COBOL e. JCLs Puoi anche esportare i cataloghi BPE e TPE, le definizioni delle risorse CICS e i dati migrati come file sequenziali e file VSAM in questo modo:

   ```
   ls  $HOME/migrated_app/*/*
   ```

4. Crea la struttura delle cartelle eseguendo lo script con il seguente comando: `create_project`

   ```
   sh $HOME/bin/create_project
   ```

5. Attiva l' CardDemo ambiente scaricando il file di `carddemo.env` installazione utilizzando:

   ```
   source $HOME/bin/carddemo.env
   ```

Fase 2: Creare una regione TPE

Dopo aver attivato l'ambiente in cui si desidera distribuire l'applicazione, è necessario creare una regione TPE.

1. Create una regione TPE utilizzando il `kixregion createRegion` comando che richiede input come, e`$KIXSYS`. `$JAVA_HOME` `$KIXLICDIR` Queste variabili di ambiente sono già configurate nel file di `carddemo.env` installazione.

```
kixregion createRegion $KIXSYS $JAVA_HOME $KIXLICDIR
```

2. Configurare la regione TPE utilizzando il `kixregion setAttr` comando.

```
kixregion setAttr $KIXSYS server.tx.languages.cobol.enabled true
kixregion setAttr $KIXSYS server.tx.languages.cobol.flavor vcobol
kixregion setAttr $KIXSYS server.tx.languages.cobol.home $VCOBOL
kixregion setAttr $KIXSYS maps.location $PROJECT_ROOT/maps
kixregion setAttr $KIXSYS programs.location $PROJECT_ROOT/loadlib
kixregion setAttr $KIXSYS environment.KIXDATA $KIXDATA
kixregion setAttr $KIXSYS td.jobq.submission.node $EBMHOME
kixregion setAttr $KIXSYS td.jobq.submission.subsys $EBMSYS
```

3. Genera il file di ambiente utente specifico per questa regione TPE eseguendo il `kixregion createScript` comando. Questo comando crea o aggiorna `$KIXSYS/bin/userenv` in base alla configurazione della regione TPE.

```
kixregion createScript $KIXSYS
```

4. Attiva la regione TPE generando il file di ambiente utente (). `$KIXSYS/bin/userenv`

```
source $KIXSYS/bin/userenv
```

5. Crea la regione TPE eseguendo il comando. `kixinstall2`

```
kixinstall2
```

Fase 3: Creare il nodo e il sottosistema BPE

Dopo aver creato una regione TPE, è necessario creare il nodo e il sottosistema BPE seguendo questi passaggi.

1. Modifica la proprietà e le autorizzazioni di. `INSTEBM`

```
sudo chown root $INSTEBM
```

```
sudo chmod 4755 $INSTEBM
```

2. Crea un nodo BPE utilizzando il `INSTEBM` comando. La directory del nodo BPE viene fornita come parametro di input.

```
$INSTEBM $EBMHOME
```

3. Attiva l'ambiente batch estraendo il `batchenv` file dal nodo BPE appena creato.

```
source $EBMHOME/batchenv
```

4. Creare il sottosistema BPE all'interno di questo nodo utilizzando Batch Administration Manager (bam). Il `bam` comando aprirà l'interfaccia di Batch Administration Manager.

```
bam
```

   a. Avviare il nodo BPE utilizzando l'interfaccia BAM. Scegliete l'opzione 2, Ambienti di sistema dal menu principale.



   b. Scegliete l'opzione 2, Start/ (Stop) Batch Node per avviare il nodo BPE.

```
                System Environments                          2024/09/17 21:27:03




             1   Report System Status
             2   Start/(Stop) Batch Node
             3   Assign the Console
             4   Change the Date
             5   Redirect Job Output
             6   Inter-Node Communications
             7   Job Accounting
             8   Assign the Initial Job Number
             9   Enable/Disable Duplicate Name Execution Delay : Enabled

             H - Help
             R - Return to Main Menu

--------------------------------------------------------------------------------
Type an option and press Return: 2
```

c. Una volta avviato, premi due volte il tasto Invio per tornare al menu principale di BAM.

```
                System Environments: Start Batch Node        2024/09/17 21:28:28









              Batch Node Startup Completed.









--------------------------------------------------------------------------------
Press Return to Continue
```

d. Per creare il sottosistema BPE, scegliete l'opzione 3, Applicazioni e sottosistemi.

```
                Batch Administration Manager                2024/09/17 21:29:03




             1   Software License Management
             2   System Environments
             3   Applications & Subsystems
             4   Security & Users
             5   Classes & Activities
             6   Problem Determination



             H - Help
             Q - Quit

--------------------------------------------------------------------------------
Type an option and press Return: 3
```

e. Quindi scegli l'opzione 3, Crea un sottosistema.

```
                Applications & Subsystems                   2024/09/17 21:29:57




             1   List All Subsystems
             2   Query a Subsystem
             3   Create a Subsystem
             4   Update a Subsystem
             5   Delete a Subsystem
             6   Change/Show Default Subsystem
             7   Import a Subsystem (from another batch node)
             8   Create a BPESUB Project
             9   Export Subsystem bldsub Config File

             H - Help
             R - Return to Main Menu

--------------------------------------------------------------------------------
Type an option and press Return: 3
```

f. Immettere il nome del sottosistema come. `sys1`

```
                Applications & Subsystems: Create          2024/09/17 21:30:22









         No Subsystems are currently defined.









             R - Return to Previous Screen

--------------------------------------------------------------------------------
Enter the Subsystem's name you want to create: sys1
```

g. Scegliete l'opzione 3, Gestione dei dati.

```
                Applications & Subsystems: Create          2024/09/17 21:30:53



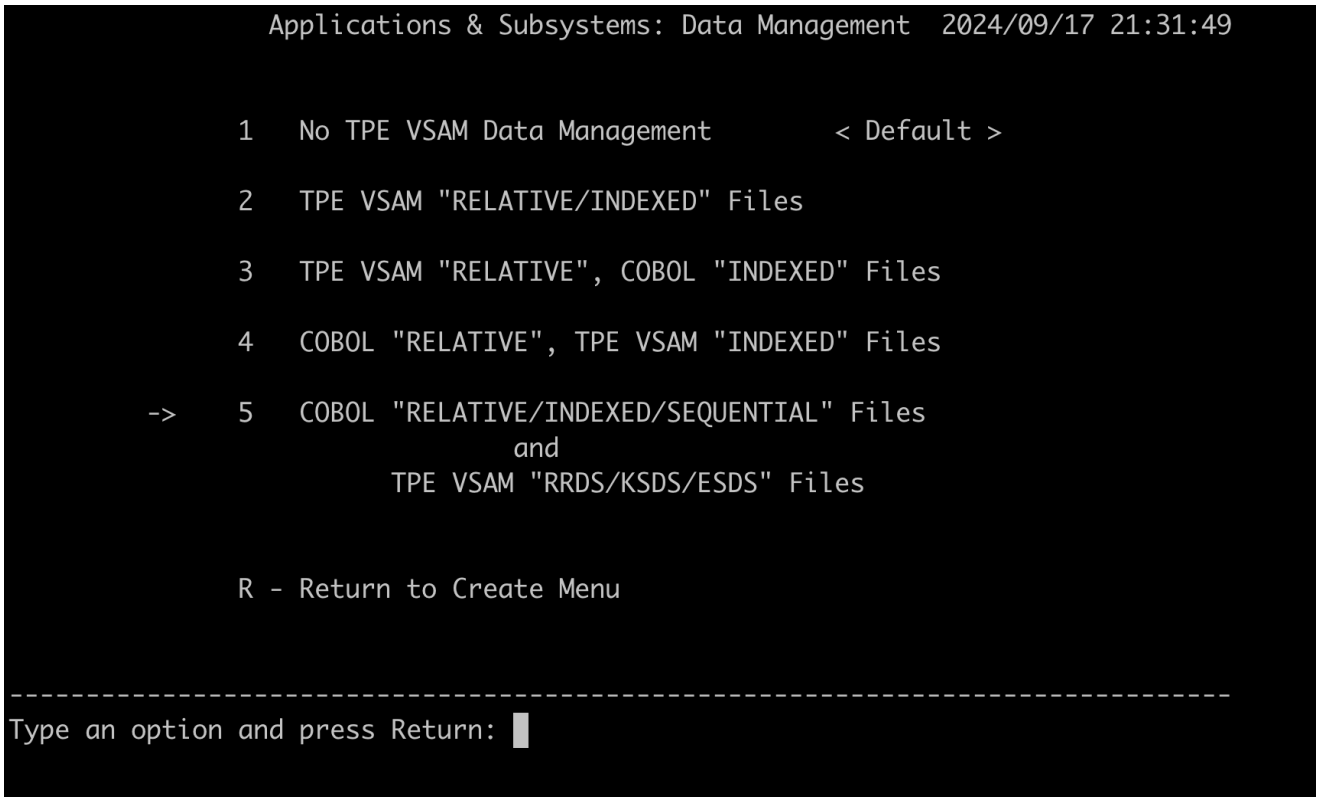
             D   Display Current sys1 Subsystem's Configuration
             S   Set to Default Subsystem Configuration

             1   Application Languages
             2   Database Management System (DBMS)
             3   Data Management
             4   Optional Packages
             5   Date/Time Management
             6   User-Specific Objects
             7   Configuration Options

             C - Create sys1 Subsystem
             H - Help
             R - Return to Main Menu

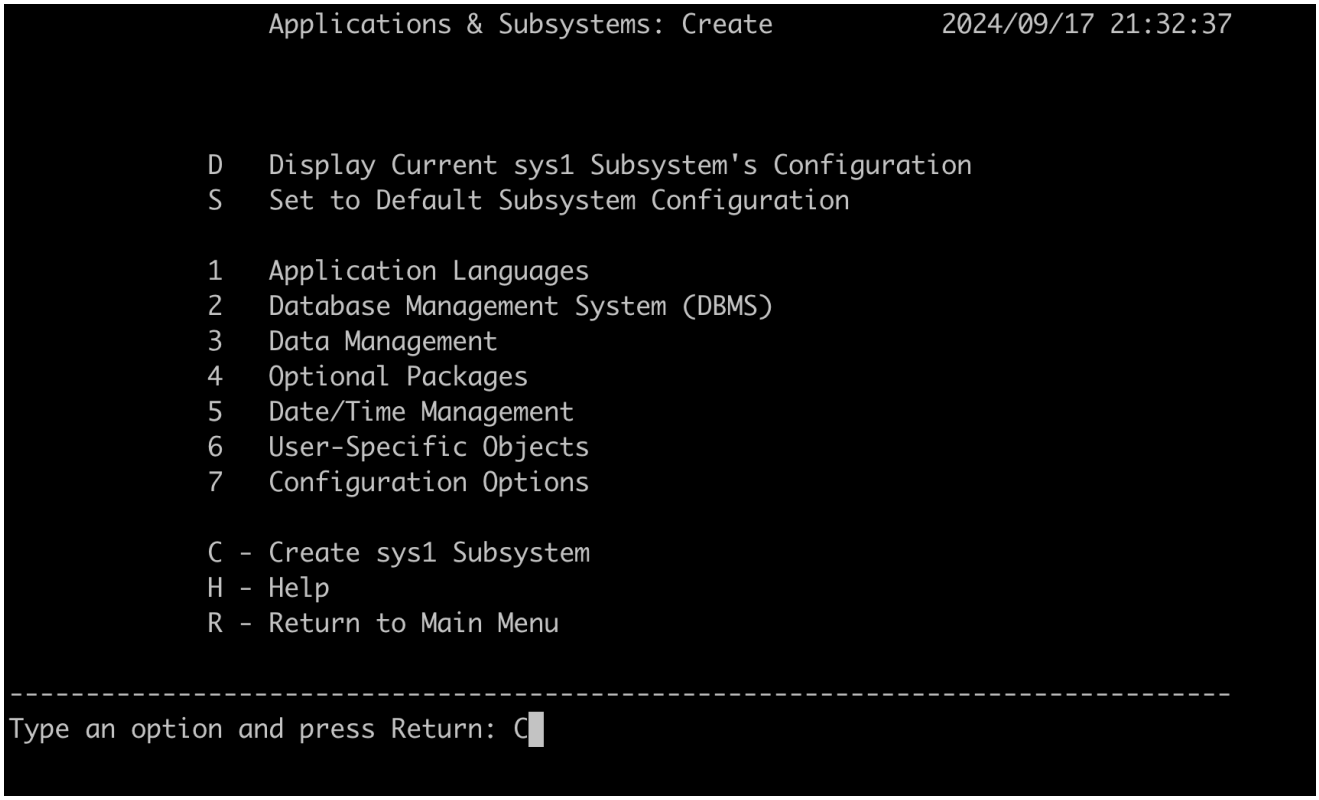
--------------------------------------------------------------------------------
Type an option and press Return: 3
```

h. Scegliete l'opzione 5, poiché l' CardDemo applicazione include sia file sequenziali che VSAM.



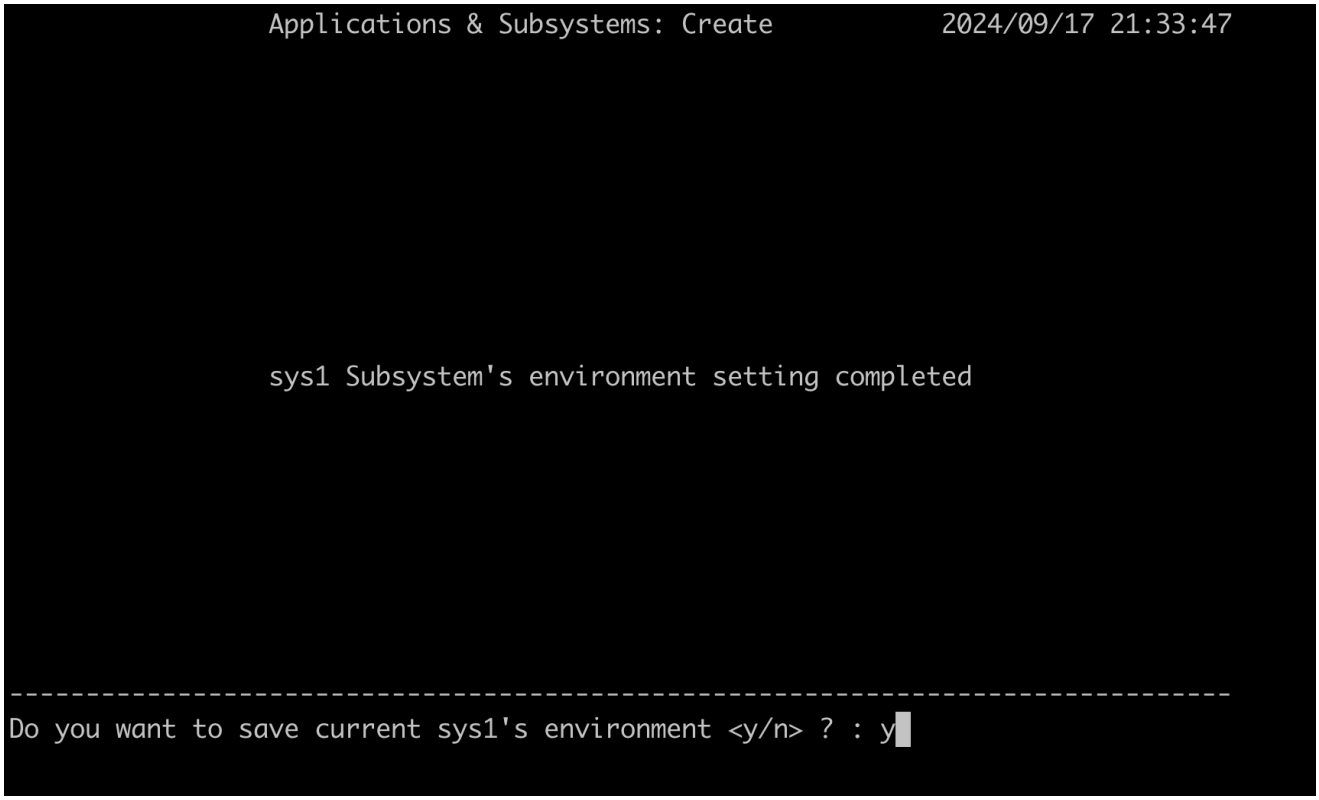
i. (Facoltativo). Premi «R» per tornare alla pagina Crea menu, rivedi le diverse opzioni di configurazione disponibili.

j. Nella pagina Crea, inserisci «C» per creare il sottosistema`sys1`.

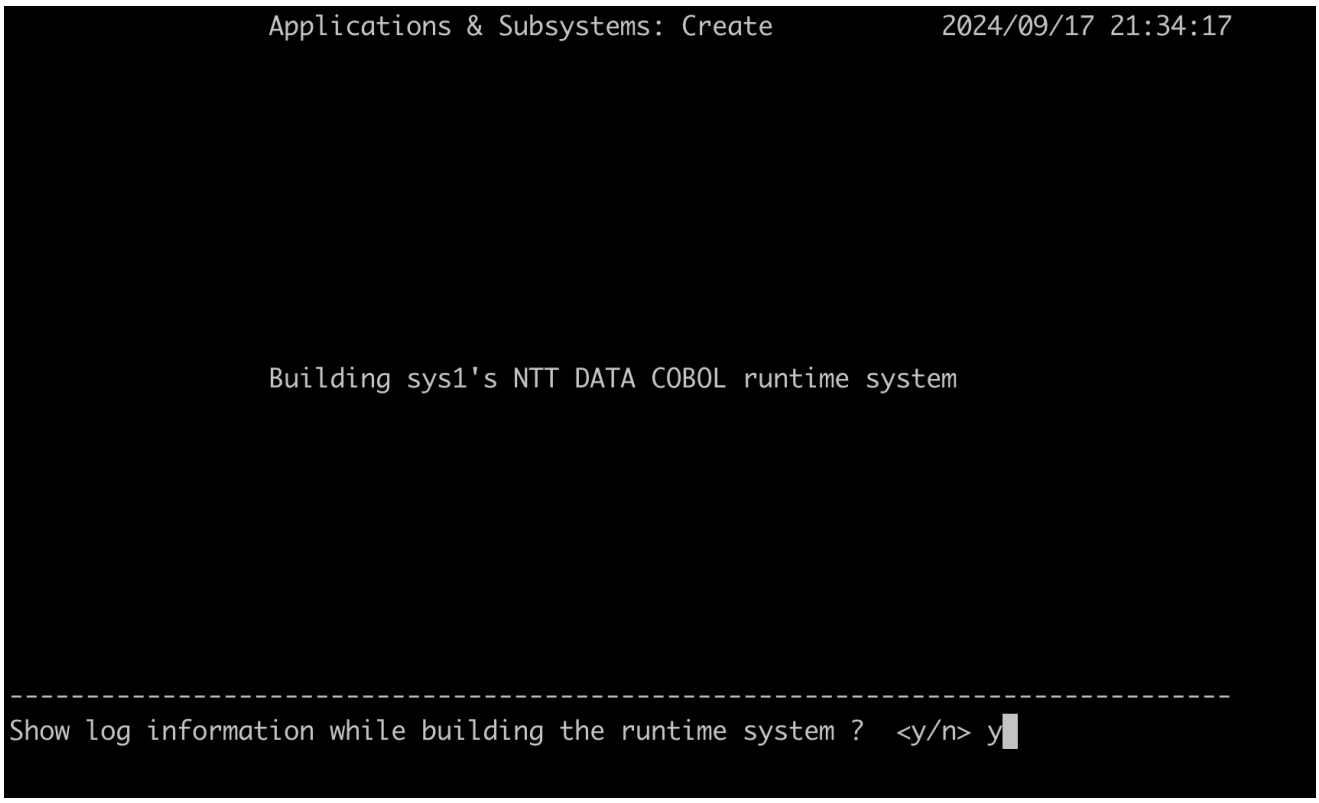


k. Rivedi le impostazioni e inserisci «C» per continuare con il resto delle impostazioni dell'ambiente. Queste impostazioni di ambiente sono precompilate grazie alle variabili di ambiente necessarie definite nel file di `carddemo.env` installazione e alla struttura di cartelle consigliata.

l. Immettere «y» per confermare e salvare le impostazioni ambientali correnti.



m. Immettete «y» per visualizzare il registro durante la creazione del sottosistema.



n. Premi il tasto Invio fino a tornare al menu principale e uscire dall'interfaccia BAM selezionando l'opzione Esci.

```
                Applications & Subsystems: Create          2024/09/17 21:43:12









                COBOL runtime system created









--------------------------------------------------------------------------------
Press Return to Continue
```

```
                Applications & Subsystems: Create          2024/09/17 21:43:55


         Subsystem sys1 created. Configuration updated.

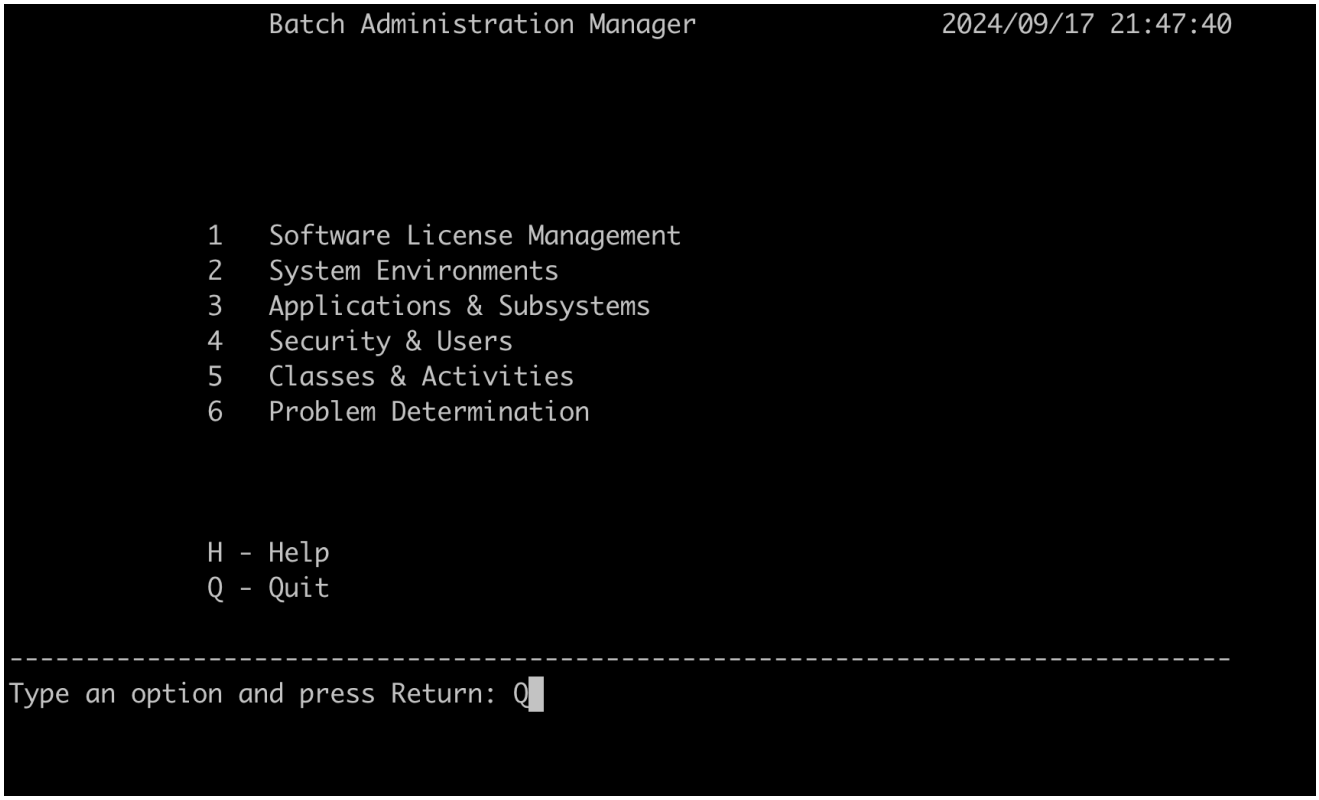
         o  To set this Subsystem's environment you must source the node
            batchenv file passing sys1 as the Subsystem argument.
            For example, after exiting BAM, type:

            .  /home/ec2-user/unikixdemo/bpenode/batchenv sys1

         o  To customize the Subsystem's environment select
            "Update a Subsystem".




--------------------------------------------------------------------------------
Press Return to Continue
```



5. Attiva il sottosistema BPE rifornendoli con il nome del sottosistema. `batchenv sys1`

```
source $EBMHOME/batchenv sys1
```

Fase 4: Compilare e distribuire l'applicazione CardDemo

In questa fase, si compilano i programmi COBOL e si distribuiscono elementi applicativi come JCL, procedure, file di dati e definizioni delle risorse CICS.

1. Attiva nuovamente l' CardDemo ambiente utilizzando il file di installazione. `carddemo.env`

```
source $HOME/bin/carddemo.env
```

2. Passate alla directory dei sorgenti COBOL.

```
cd $MIGAPP_DIR/cbl
```

3. Compila il programma `CBACT01C.cbl` Cobol usando uno script. `compile`

```
compile CBACT01C.cbl
```

4. Compila tutti i programmi Cobol usando uno script. `compile.all`

```
compile.all
```

5. Vai alla cartella dei sorgenti delle mappe BMS.

```
cd $MIGAPP_DIR/bms
```

6. Compila la mappa `COACTUP.bms` BMS usando lo script. `compbms`

```
compbms COACTUP.bms
```

7. Compila tutte le mappe BMS usando lo script. `compbms.all`

```
compbms.all
```

8. Verifica i binari compilati per le mappe COBOL e BMS.

```
ls $PROJECT_ROOT/loadlib
ls $PROJECT_ROOT/maps
```

9. Implementa altri elementi dell'applicazione come JCL, procedure, file di dati e definizioni di risorse CICS utilizzando lo script. `deploy_app`

```
deploy_app
```

10. Passa alla directory JCL del progetto.

```
cd $PROJECT_ROOT/jcl
```

11. Traduci JCL ACCTFILE in BPE JCL Macro. Usa il `mvstrans` comando, utilizzando l'opzione «-v» per la verifica JCL e l'opzione «-f» per creare la macro.

```
mvstrans ACCTFILE -v
mvstrans ACCTFILE -f
```

12. Traduci la procedura JCL REPROC in una macro della procedura JCL BPE. Usa il `mvstrans` comando con l'opzione «-p» insieme all'opzione «-v» per la verifica e l'opzione «-f» per creare la macro.

```
mvstrans REPROC -v -p
```

```
mvstrans REPROC -f -p
```

13. Translate tutte le procedure JCLs e JCL.

```
for file in "./jmvs/*"; do mvstrans $file -f; done > jmvs.out
for file in "./mvsp/*"; do mvstrans $file -p -f; done > mvsp.out
```

Fase 5: Importazione del catalogo BPE e TPE

In questo passaggio, si importa il catalogo BPE e TPE utilizzando comandi diversi.

1. Importa il catalogo BPE utilizzando il comando. `loadcat`

```
loadcat $MIGAPP_DIR/catlg/bpe/BPECAT*
```

2. Vai alla directory $KIXSYS.

```
cd $KIXSYS
```

3. Importa il catalogo TPE usando il comando. `kiximpcat`

```
kiximpcat -c CATALOG -l CATALOG.lst
```

4. Importa le definizioni delle risorse CICS usando il comando kiximptbl.

```
kiximptbl
```

Fase 6: Avviare e collegare TPE con BPE

In questo passaggio, è necessario avviare la regione TPE creata in precedenza insieme al gestore BPE e collegarli per poter eseguire l'applicazione di esempio. CardDemo

1. Eseguite il `kixverify` comando su tutti i file VSAM per assicurarvi che vengano ripristinati e che tutti i file precedentemente aperti vengano chiusi.

```
kixverify -r ALL
```

2. Avviate la regione TPE.

```
kixregion start $KIXSYS
```

3. Assicuratevi che sia BPE che TPE siano collegati. Questo è fondamentale in quanto i file VSAM sono di proprietà di TPE e qualsiasi operazione in batch che accede al VSAM richiederà una connessione a TPE.

```
ebmsys -t
```



Fase 7: Eseguire l'applicazione CardDemo

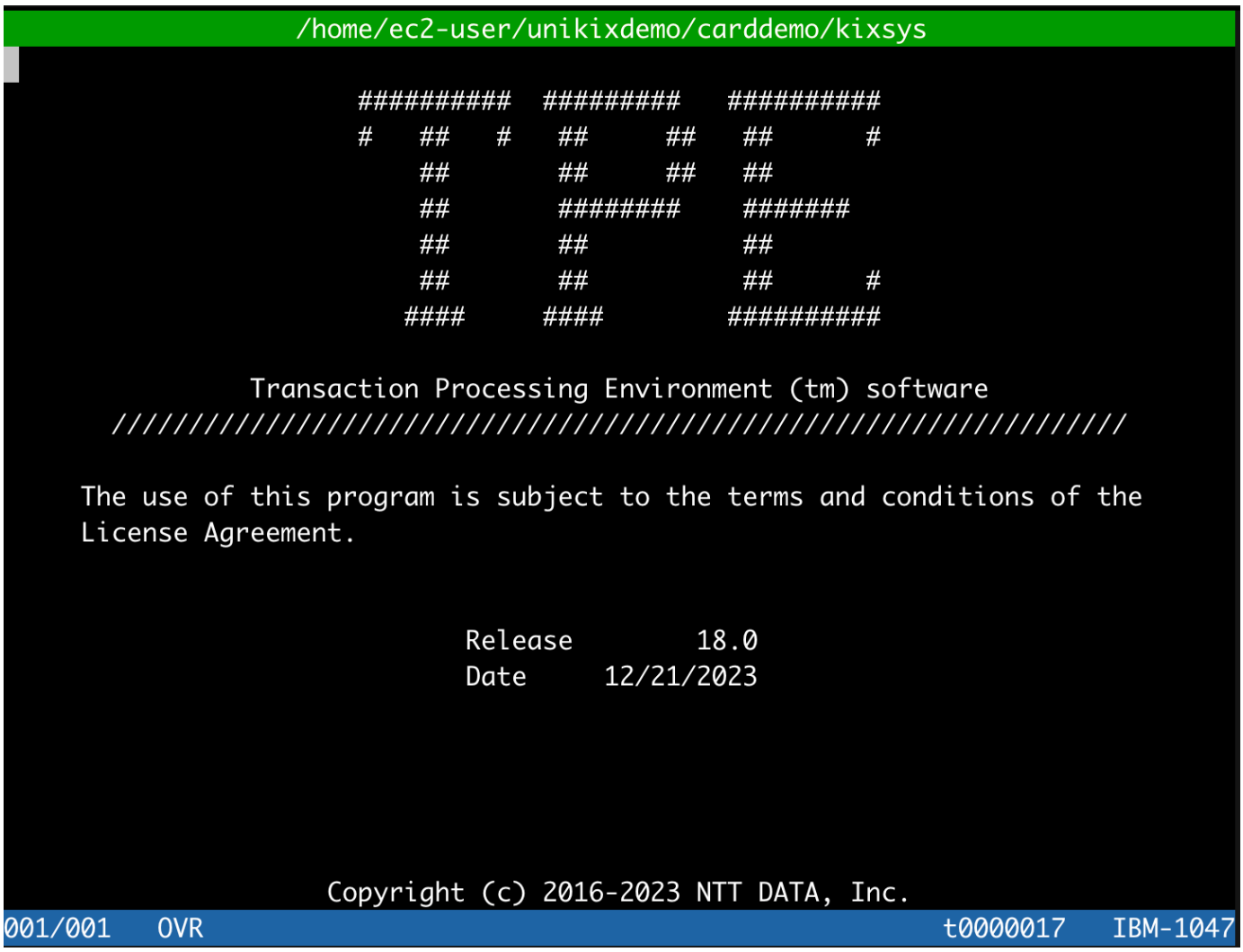
In questo passaggio, si esegue l' CardDemo applicazione nell'emulatore di TN3270 terminale.

L'AMI UniKix runtime include l'emulatore di terminale TN3270 che puoi avviare direttamente dall'istanza EC2. UniKix

Connect a TPE utilizzando l'emulatore di TN3270 terminale

- Avvia il TN3270 terminale usando il comando. `kixterm`

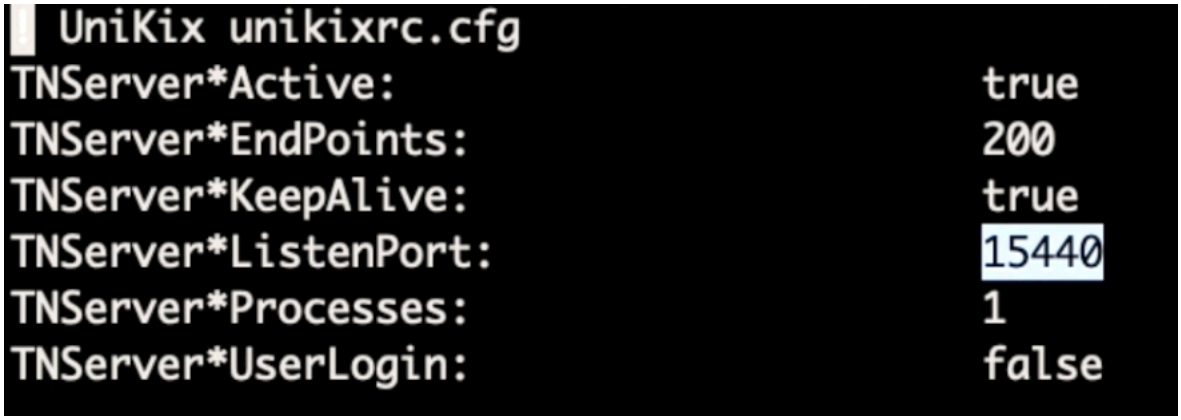
```
kixterm
```



(Facoltativo). Se vuoi usare il tuo emulatore di terminale:

1. Ottieni l'indirizzo IP dell'istanza di UniKix runtime dalla console Amazon EC2.
2. Ottieni il numero di porta per la connessione alla regione TPE utilizzando l'emulatore di TN3270 terminale. Puoi trovarlo nel TNServer ListenPort file unikixrc.cfg.

```
cat $KIXSYS/unikixrc.cfg
```

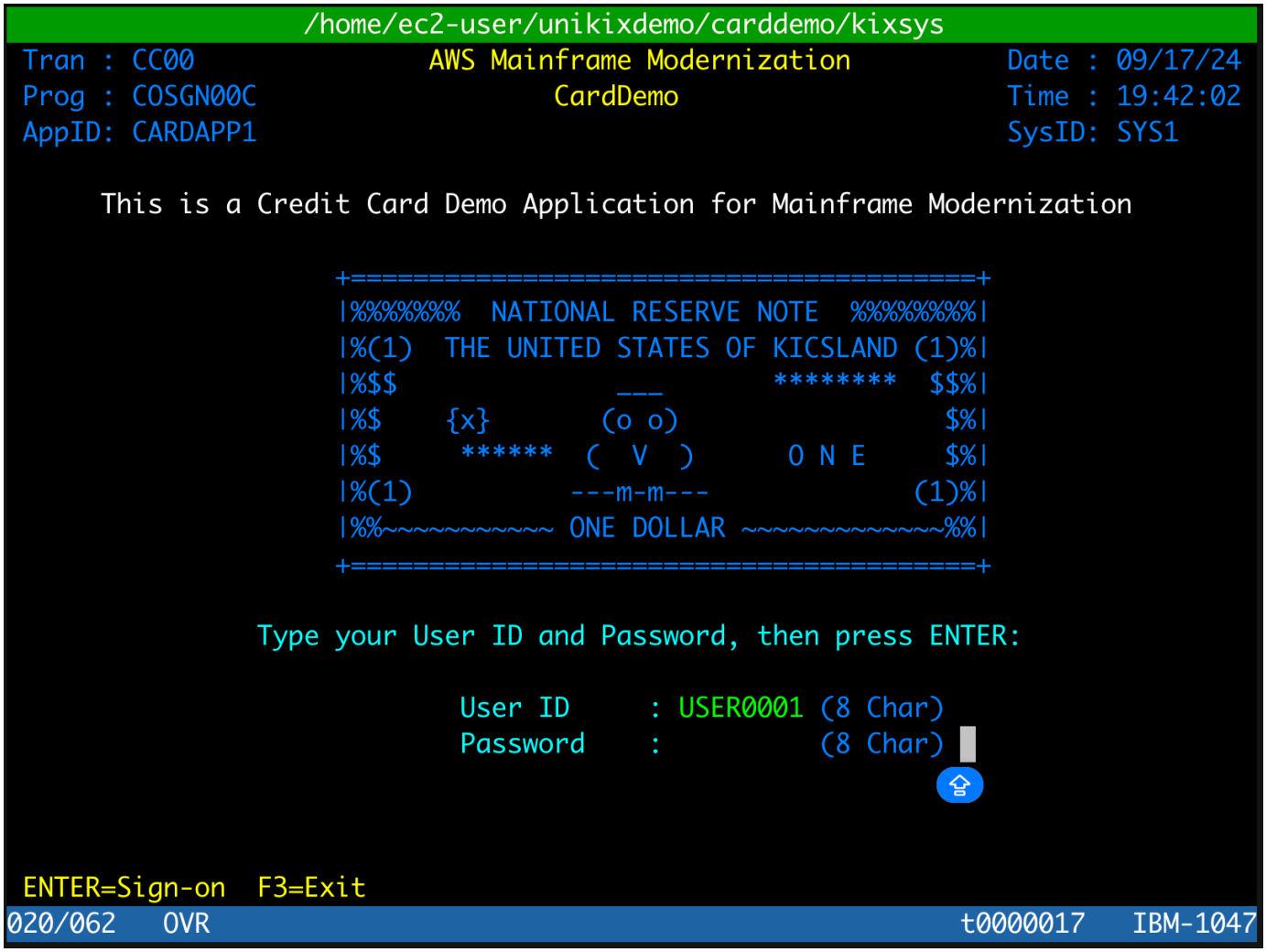


3. Configura l'emulatore di TN3270 terminale per utilizzare l'indirizzo IP dell'istanza di UniKix runtime e il numero di porta 15440.

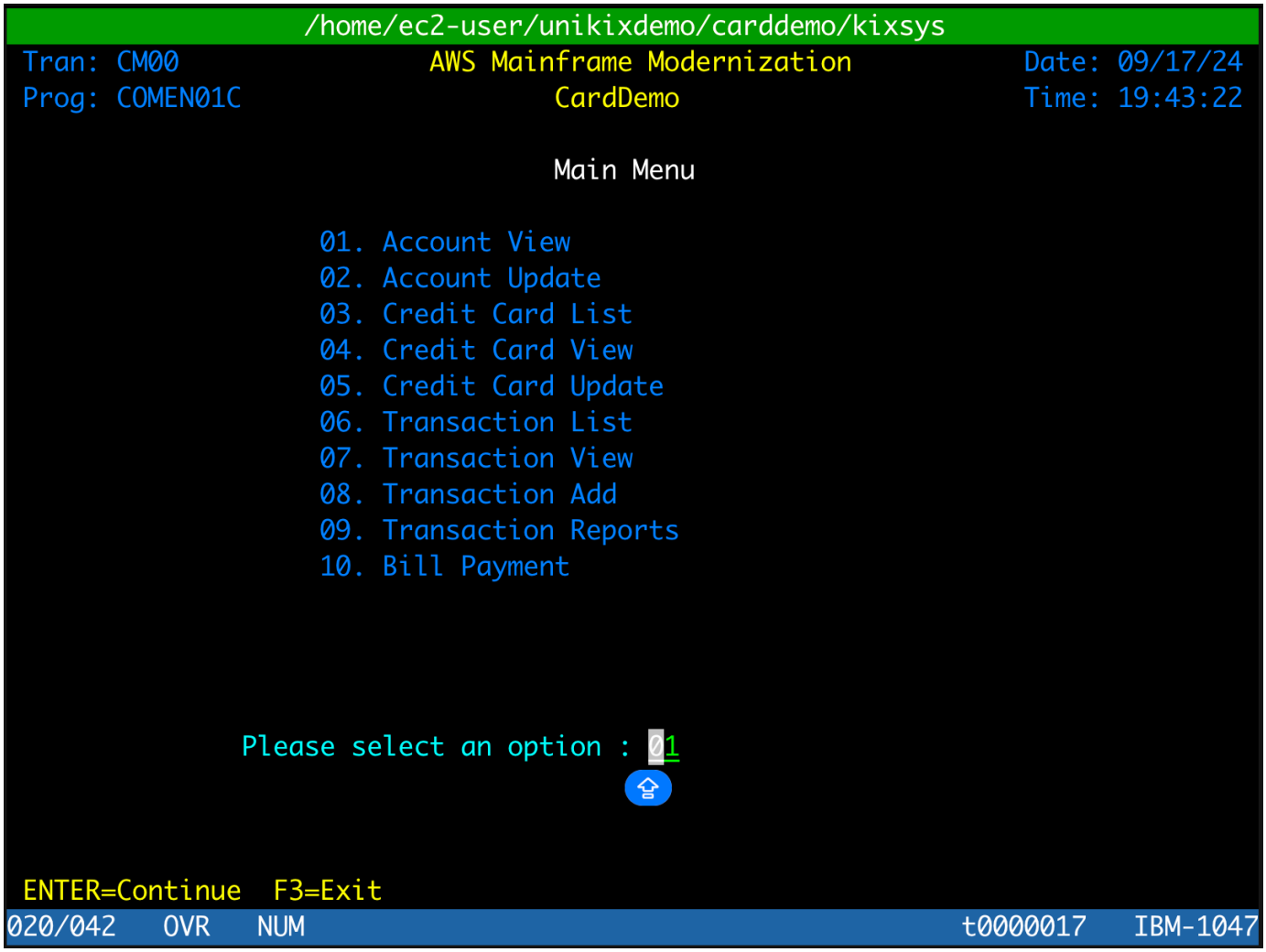
Transazioni online

Questa sezione presuppone che l'utente si sia connesso all'emulatore di TN3270 terminale utilizzando il `kixterm` comando.

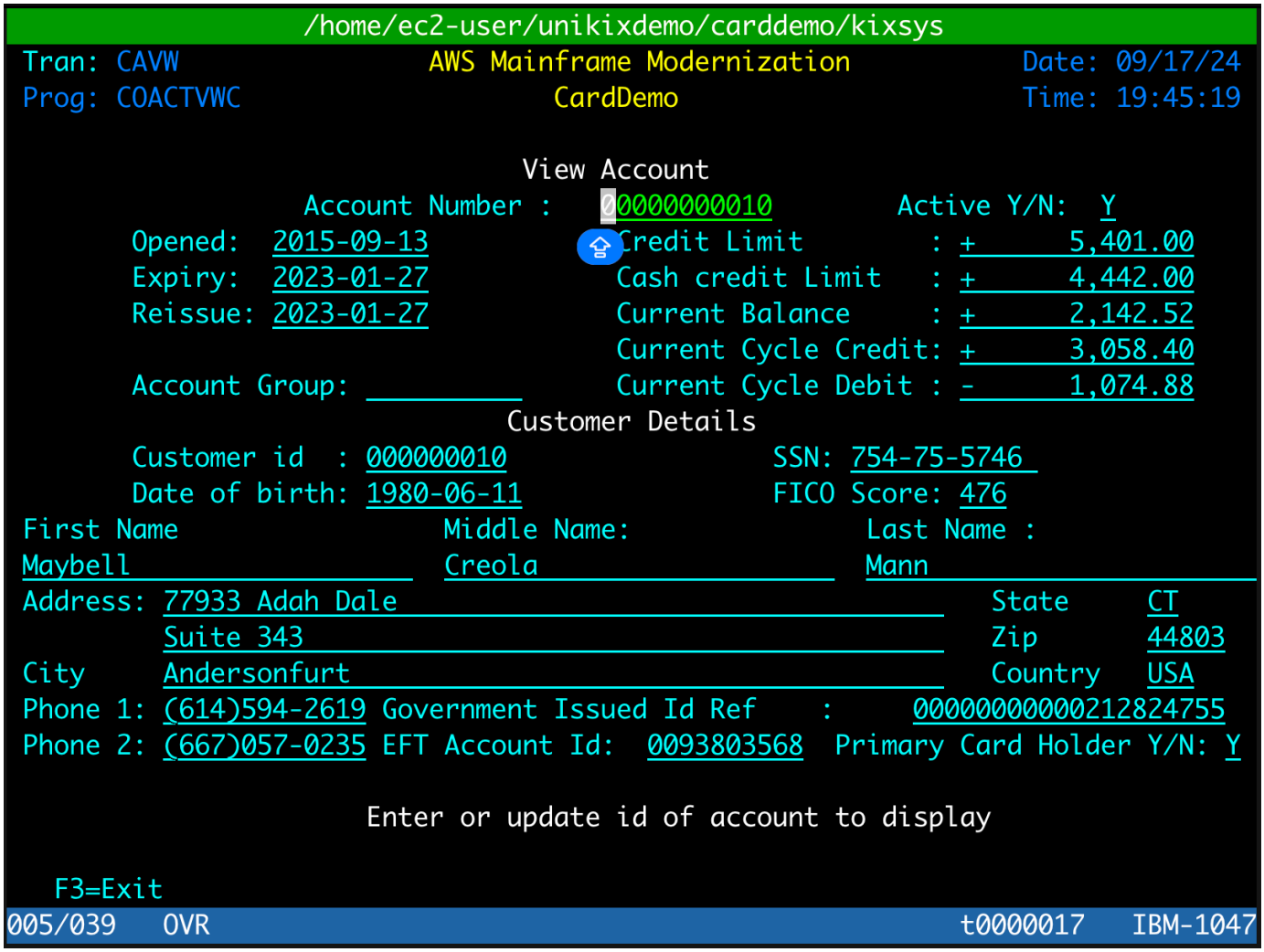
1. Dopo la connessione dall'emulatore di TN3270 terminale, premi il tasto «Invio» per cancellare la schermata TPE e inserire la transazione iniziale.
2. Nella transazione iniziale CC00 (schermata di accesso) inserisci il nome `USER001` utente e `PASSWORD` la password.



3. Scegli l'opzione «01" dal menu principale per visualizzare gli account.



4. Nella schermata Visualizza account, inserisci un numero di conto (ad esempio, 00000000010). Dovresti vedere le informazioni sull'account compilate con i dati migrati.



5. Premi due volte il tasto «PF03» per tornare alla schermata di accesso e esci dal TN3270 terminale premendo «Ctrl+C» (Windows) o «Cmd+C» (Macbook).

Lavori in batch

1. Vai alla directory JCL.

```
cd $MBMSUB
```

2. Invia il lavoro `MFCATGL1` e osserva l'output del registro del lavoro.

```
BPESUB READCARD
```

3. Facoltativamente, è possibile visualizzare i registri dei lavori dalla `$SUBSYS_OUTDIR` directory.

```
ls -lrt $SUBSYS_OUTDIR/*
```

Ora avete distribuito con successo l' CardDemo applicazione sul UniKix runtime NTT DATA e verificato l'applicazione in esecuzione navigando tra alcune schermate online CICS e i processi in batch.

Risoluzione dei problemi

Di seguito sono riportati alcuni errori comuni che è possibile riscontrare durante la configurazione dell'applicazione. CardDemo

Errore: errore di licenza

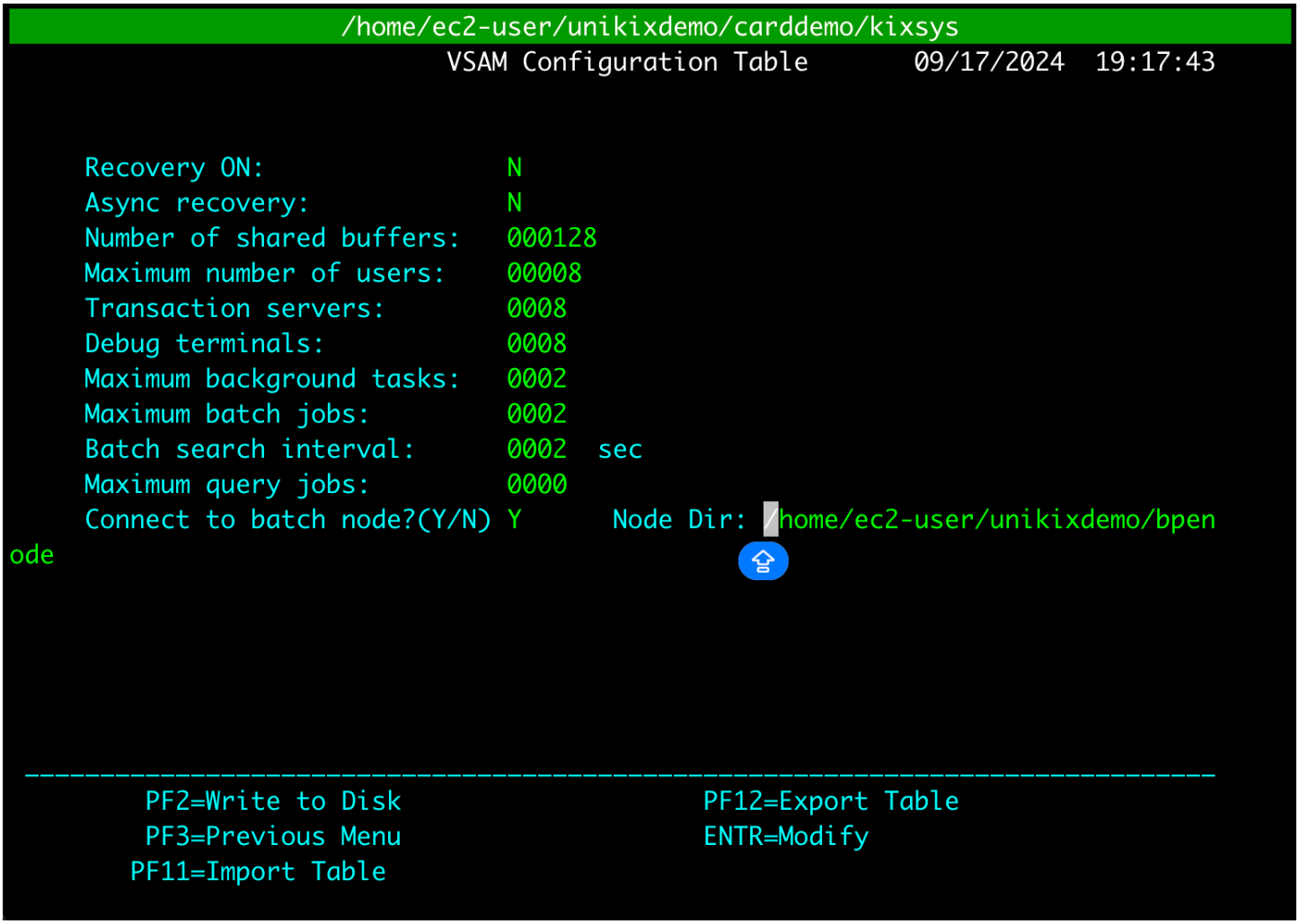
Se ricevi un errore di licenza mentre segui questo tutorial, è possibile che IMDSv2sia abilitato nel tuo EC2. Puoi risolvere questo problema modificando l'opzione dei metadati dell'istanza su Opzionale, come indicato in Restore the use of IMDSv1 nella guida IMDSv2per l'utente di Amazon EC2.

Errore: TPE non è collegato a BPE

Se il TPE non è collegato a BPE, assicurati che la tabella di configurazione VSAM sia configurata correttamente con la directory BPE Node. Per accedere alla tabella di configurazione VSAM, avvia l'emulatore di TN3270 terminale utilizzando il seguente comando:

```
kixterm
```

1. Inserisci il nome della transazione come. `CTBL`
2. Nel menu Table Manager, scegliete l'opzione Tabelle standard.
3. Nella schermata Tabelle standard, scegli l'opzione Tabella di configurazione VSAM.
4. Controlla se Connect to batch node? è impostato su "Y» e la cartella dei nodi è corretta.


/home/ec2-user/unikixdemo/carddemo/kixsys
VSAM Configuration Table 09/17/2024 19:17:43
Recovery ON: N
Async recovery: N
Number of shared buffers: 000128
Maximum number of users: 00008
Transaction servers: 0008
Debug terminals: 0008
Maximum background tasks: 0002
Maximum batch jobs: 0002
Batch search interval: 0002 sec
Maximum query jobs: 0000
Connect to batch node?(Y/N) Y Node Dir: /home/ec2-user/unikixdemo/bpen
ode
PF2=Write to Disk
PF3=Previous Menu
PF11=Import Table
PF12=Export Table
ENTR=Modify

Sicurezza nella modernizzazione dei AWS mainframe

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . 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 AWS servizio 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 aiuta a capire come applicare il modello di responsabilità condivisa quando si utilizza la modernizzazione del AWS mainframe. 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.

AWS Mainframe 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.

AWS La 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 dei AWS mainframe
- Identity and Access Management per la modernizzazione del AWS mainframe
- Convalida della conformità per la modernizzazione del mainframe AWS
- Resilienza nella modernizzazione dei AWS mainframe
- Sicurezza dell'infrastruttura in Modernizzazione del mainframe AWS
- Accesso Modernizzazione del mainframe AWS tramite un endpoint di AWS PrivateLink interfaccia

Protezione dei dati nella modernizzazione dei AWS mainframe

Il modello di responsabilità AWS condivisa modello di di si applica alla protezione dei dati nella modernizzazione del AWS mainframe. 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. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per maggiori informazioni sulla privacy dei dati, consulta le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al AWS Modello di responsabilità condivisa e GDPR nel AWS Blog sulla sicurezza.

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. Suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- SSL/TLS Da utilizzare per comunicare con AWS le risorse. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta Lavorare con i CloudTrail percorsi nella Guida per l'AWS CloudTrail utente.
- 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 140-3 per accedere AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-3.

Ti consigliamo 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. AWS CLI AWS SDKs 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 si fornisce un URL a un server esterno, suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la richiesta al server.

Dati raccolti da AWS Mainframe 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 Gestione dei segreti AWS 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 WorkSpaces Applications, 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 distribuisci 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 Amazon Elastic File System o Amazon FSx file system 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 FSx crittografia Amazon EFS e Amazon se colleghi uno o più di questi file system al tuo ambiente di runtime.

Crittografia dei dati inattiva per il servizio di modernizzazione AWS 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 Customer managed keys 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 con chiavi gestite dal cliente
`Definition` Contiene la definizione per una particolare applicazione.	Abilitato	Abilitato
`EnvironmentSummary`	Abilitato	Abilitato

Tipo di dati	Chiave gestita da AWS crittografia	Crittografia con chiavi gestite dal cliente
Contiene informazioni sull'ambiente di runtime.		
`ApplicationSummary` Contiene informazioni sull'applicazione AWS Mainframe Modernization.	Abilitato	Abilitato
`DeploymentSummary` 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 AWS KMS costi. 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:

- DescribeKeyInviate 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 GenerateDataKeyrichieste 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 ridistribuzione 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
  - Crittografa
  - ReEncryptFrom
  - ReEncryptTo
  - GenerateDataKey
  - DescribeKey
  - CreateGrant
- Creare o ridistribuire 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 Transform for mainframe o Rocket Software (precedentemente 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 distribuisci un'applicazione, sia i dati che i dati di servizio vengono scaricati su volumi Amazon EBS. Se scegli di collegare FSx lo storage Amazon EFS o Amazon 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.

Creazione di una chiave gestita dal cliente

È possibile creare una chiave simmetrica gestita dal cliente utilizzando Console di gestione AWS o il. AWS KMS APIs

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 della chiave

Le policy della chiave controllano l'accesso alla chiave gestita dal cliente. Ogni chiave gestita dal cliente deve avere esattamente una policy della chiave, che contiene istruzioni che determinano chi

può usare la chiave e come la possono usare. Quando crei la chiave gestita dal cliente, è possibile specificare una policy della 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 delle concessioni, consulta la Guida per gli sviluppatori di AWS Key Management Service .

  Ciò consente a AWS Mainframe Modernization di eseguire le seguenti operazioni:

  - Chiama `GenerateDataKey` per generare una chiave dati crittografata e archiviarla, poiché la chiave dati non viene utilizzata immediatamente per crittografare.
  - Chiama `Decrypt` per utilizzare la chiave dati crittografata memorizzata per accedere ai dati crittografati.
  - Configurare un principale ritirato per consentire al servizio di `RetireGrant`.
- `kms:DescribeKey`— Fornisce dettagli chiave gestiti dal cliente per consentire a AWS Mainframe Modernization 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM User Permissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333: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. Immetti 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 Console di gestione AWS o il. AWS CLI

Per specificare la chiave gestita dal cliente durante la creazione di un ambiente di runtime in Console di gestione AWS, vedereCrea un ambiente di runtime per la modernizzazione del AWS mainframe. Per specificare la chiave gestita dal cliente durante la creazione di un'applicazione in Console di gestione AWS, vedereCrea un'applicazione 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": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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": "AROA3YPCLM65MZFUPM4JO: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": "AROAIGDTESTANDEXAMPLE: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": "AROA3YPCLM65EEXVIEH7D:AutoScaling",
        "arn": "arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForAutoScaling/
AutoScaling",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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'`Decrypt`operazione 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": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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": "AROA3YPCLM65CLCEKKC7Z: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": "AROAIGDTESTANDEXAMPLE: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": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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'`Decrypt`operazione 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": "AROA3YPCLM65BSPZ37E6G:m2-hm-bqe367dxtfcpdbzmnhfzranisu",
        "arn": "arn:aws:sts::111122223333:assumed-role/
SupernovaEnvironmentInstanceScopeDownRole/m2-hm-bqe367dxtfcpdbzmnhfzranisu",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE: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 dei dati 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 for protocol del servizio di modernizzazione del AWS mainframe non sono crittografati durante il transito. TN3270 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 il servizio. APIs 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 un sistema Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. 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 tramite 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 Modernizzazione del mainframe AWS

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi relativi alla modernizzazione del AWS mainframe, all'identità e all'accesso.)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta Come funziona la modernizzazione AWS del mainframe con IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta Esempi di policy basate sull'identità per la modernizzazione del mainframe AWS)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta Che cos'è il Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la

federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come

creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno 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 AWS mainframe.

Funzionalità IAM che puoi utilizzare con Mainframe Modernization AWS

Funzionalità IAM	AWS Supporto per la modernizzazione del mainframe
Policy basate sull'identità	Sì
Policy basate su risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	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 sull'identità: sì

Le policy basate sull'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 operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente di IAM.

Con le policy basate sull'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. 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 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 collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio specificare un'entità principale. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Azioni politiche per AWS la modernizzazione del mainframe

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere 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 azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `List`, includi la seguente azione:

```
"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 relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità 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'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon

(ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si 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 utilizzandole ARNs per identificare la risorsa a cui si applica la policy IAM. Per ulteriori informazioni sul formato di ARNs, consulta Amazon Resource Names (ARNs) nel 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 un elenco dei tipi di risorse per la modernizzazione del AWS mainframe e relativi tipi di risorse ARNs, 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 (politiche 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 azioni corrispondenti per le quali è possibile concedere le autorizzazioni per eseguire l'azione.
- 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 modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources
CancelBatchJobExecution		Applicazione

CreateApplication	`iam:PassRole`
	
	`kms:DescribeKey`
		Applicazione

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources

	`kms:CreateGrant`
	
	`s3:GetObject`
	
	`s3:ListBucket`	

CreateDataSetImportTask	`s3:GetObject`	Applicazione

CreateDataSetExportTask	`kms:DescribeKey`
	
	`s3:PutObject`
		Applicazione

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources
CreateDeployment	elasticloadbalancing:AddTags elasticloadbalancing:CreateListener elasticloadbalancing:CreateTargetGroup elasticloadbalancing:RegisterTargets elasticloadbalancing:DeleteListener elasticloadbalancing:DeleteTargetGroup elasticloadbalancing:DeregisterTargets elasticloadbalancing:DeleteLoadBalancer logs:CreateLogDelivery logs:GetLogDelivery logs:UpdateLogDelivery logs:DeleteLogDelivery logs:ListLogDeliveries logs:PutResourcePolicy	Applicazione

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources

	`logs:DescribeResou rcePolicies`
	
	`logs:DescribeLogGroups`	

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources

CreateEnvironment	`ec2:CreateNetworkInterface`
	
	`ec2:CreateNetworkI nterfacePermission`
	
	`ec2:DescribeNetwor kInterfaces`
	
	`ec2:DescribeSecurityGroups`
	
	`ec2:DescribeSubnets`
	
	`ec2:DescribeVpcAttribute`
	
	`ec2:DescribeVpcs`
	
	`ec2:ModifyNetworkI nterfaceAttribute`
	
	`elasticfilesystem: DescribeMountTargets`
	
	`elasticloadbalanci ng:AddTags`
	
	`elasticloadbalanci ng:CreateLoadBalancer`
	
	`elasticloadbalanci ng:DeleteLoadBalancer`
	
	`kms:DescribeKey`
	
	`kms:CreateGrant`
	
	`fsx:DescribeFileSystems`
		Ambiente

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources
	`iam:CreateServiceL inkedRole`	

DeleteApplication	`elasticloadbalanci ng:DeleteListener`
	`elasticloadbalanci ng:DeleteTargetGroup`
	`logs:DeleteLogDelivery`
		Applicazione

DeleteApplicationF romEnvironment	`elasticloadbalanci ng:DeleteListener` `elasticloadbalanci ng:DeleteTargetGroup`	Applicazione Ambiente
DeleteEnvironment	`elasticloadbalanci ng:DeleteLoadBalancer`	Ambiente
GetApplication		Applicazione
GetApplicationVersion		Applicazione
GetBatchJobExecution		Applicazione
GetDataSetDetails		Applicazione
GetDataSetImportTa sk		Applicazione
GetDataSetExportTa sk		Applicazione
GetDeployment		Applicazione
GetEnvironment		Ambiente

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources
ListApplications		*
ListApplicationVer sions		*
ListBatchJobDefini tions		*
ListBatchJobExecut ions		*
ListDataSetImportH istory		*
ListDataSetExportH istory		*
ListDataSets		*
ListDeployments		*
ListEngineVersions		*
ListEnvironments		*
ListTagsForResource		*
StartApplication		Applicazione
StartBatchJob		Applicazione
StopApplication		Applicazione
TagResource		*
UntagResource		*

AWS Operazioni delle API di modernizz azione del mainframe	Autorizzazioni necessarie (azioni API)	Resources

UpdateApplication	`s3:GetObject`
	
	`s3:ListBucket`
		Applicazione

UpdateEnvironment	`kms:DescribeKey`	Ambiente

Le condizioni fondamentali delle policy per AWS la modernizzazione del mainframe

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile 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. 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 () in Mainframe Modernization ACLs AWS

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle 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 degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire 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 Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione 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

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS compatibili con IAM nella Guida per l'utente IAM.

Sessioni di accesso diretto per AWS la modernizzazione del mainframe

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale chiamante an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta 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 Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

AWS La modernizzazione del mainframe supporta i ruoli di servizio per i nodi di attività (transazioni, lavori, fine o completamento, ecc.).

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere la funzionalità di modernizzazione del mainframe. AWS Modifica 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'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati in Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, 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 Modernizzazione del mainframe AWS

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 sono autorizzati a creare o modificare le risorse di modernizzazione del mainframe. AWS Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) 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 di ARNs per ogni 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 AWS le risorse di modernizzazione del mainframe nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire 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 maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori 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 ad azioni e risorse è possibile aggiungere una condizione alle 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 CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM 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, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API 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. AWS Al contrario, è opportuno concedere l'accesso solo alle azioni 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 maggiori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di 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 collegate 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 AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "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:PassRole`azione, 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 trasmettere 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 i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per maggiori 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 IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente di IAM.

Utilizzo di ruoli collegati ai servizi per Modernizzazione del mainframe AWS

Modernizzazione del mainframe AWS utilizza ruoli AWS Identity and Access Management collegati ai servizi (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. Modernizzazione del mainframe AWS I ruoli collegati ai servizi sono predefiniti Modernizzazione del mainframe AWS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione Modernizzazione del mainframe AWS perché non è necessario aggiungere manualmente le autorizzazioni necessarie. Modernizzazione del mainframe AWS definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. Modernizzazione del mainframe AWS 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 al servizio solo dopo avere eliminato le risorse correlate. In questo modo proteggi Modernizzazione del mainframe AWS le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta AWS Servizi compatibili con IAM e cerca i servizi con Sì nella colonna 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 Modernizzazione del mainframe AWS

Modernizzazione del mainframe AWS utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAWSM2: configura la rete per connettersi al tuo VPC e accedere a risorse come i file system.

Il ruolo AWSService RoleFor AWSM2 collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `m2.amazonaws.com`

La politica di autorizzazione dei ruoli denominata AWSM2 ServicePolicy consente di Modernizzazione del mainframe AWS completare le seguenti azioni sulle risorse specificate:

- Crea, elimina, descrivi e collega le autorizzazioni alle interfacce di rete Amazon EC2 per Modernizzazione del mainframe AWS l'ambiente 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' Modernizzazione del mainframe AWS ambiente.
- Descrivi il FSx file system Amazon EFS o Amazon, se utilizzato.
- Invia i parametri al cliente CloudWatch dall'ambiente di runtime.

JSON

```
{
 "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 al servizio è necessario configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente IAM.

Creazione di un ruolo collegato al servizio per Modernizzazione del mainframe AWS

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente di runtime in Console di gestione AWS, the o nell' AWS API AWS CLI, Modernizzazione del mainframe AWS crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un ambiente di runtime, Modernizzazione del mainframe AWS crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato al servizio per Modernizzazione del mainframe AWS

Modernizzazione del mainframe AWS non consente di modificare il ruolo collegato al AWSService RoleFor AWSM2 servizio. Dopo avere creato un ruolo collegato al servizio, non sarà possibile 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 al servizio nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato al servizio per Modernizzazione del mainframe AWS

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 Modernizzazione del mainframe AWS servizio 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 Modernizzazione del mainframe AWS le risorse utilizzate da AWSService RoleFor AWSM2

- Eliminare gli ambienti di runtime in Modernizzazione del mainframe AWS. 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 IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al AWSService RoleFor AWSM2 servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato ai servizi nella Guida per l'utente IAM.

Regioni supportate per i ruoli collegati ai servizi Modernizzazione del mainframe AWS

Modernizzazione del mainframe AWS 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 modernizzazione del mainframe AWS

I revisori di terze parti valutano la sicurezza e la conformità della modernizzazione del AWS mainframe come parte di più AWS programmi di conformità. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco di AWS servizi nell'ambito di programmi di conformità specifici, consulta Servizi AWS nell'ambito del programma di conformità . Per informazioni generali, consulta Programmi per la conformità di AWS.

Puoi scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità in materia di conformità quando utilizzate AWS Mainframe Modernization è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Le guide rapide per la sicurezza e la conformità: queste guide all'implementazione illustrano considerazioni architettoniche e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive come le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- AWS Risorse per la conformità Risorse per la conformità: questa raccolta di potrebbe riguardare il settore e la località in cui operate.
- Evaluating Resources with Rules nella AWS Config Developer Guide: AWS Config valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- AWS Security Hub CSPM— Questo AWS servizio offre una visione completa dello stato di sicurezza dell'utente e consente di verificare la conformità agli standard e alle best practice del settore della sicurezza. AWS

Resilienza nella modernizzazione dei AWS mainframe

L'infrastruttura AWS globale è costruita attorno AWS a regioni 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 tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Infrastruttura AWS globale.

Sicurezza dell'infrastruttura in Modernizzazione del mainframe AWS

In quanto servizio gestito, Modernizzazione del mainframe AWS è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzate chiamate API AWS pubblicate per accedere Modernizzazione del mainframe AWS attraverso la rete. I client 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.

Accesso Modernizzazione del mainframe AWS tramite un endpoint di AWS PrivateLink interfaccia

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e. Modernizzazione del mainframe AWS Puoi accedere Modernizzazione del mainframe AWS come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. Modernizzazione del mainframe AWS

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Queste sono interfacce di rete gestite dal richiedente che fungono da punto di ingresso per il traffico destinato a Modernizzazione del mainframe AWS.

Per ulteriori informazioni, consulta Access Servizi AWS through AWS PrivateLink nella AWS PrivateLink Guida.

Considerazioni per Modernizzazione del mainframe AWS

Prima di configurare un endpoint di interfaccia per Modernizzazione del mainframe AWS, consulta le considerazioni nella Guida.AWS PrivateLink

Modernizzazione del mainframe AWS supporta l'effettuazione di chiamate a tutte le sue azioni API tramite l'endpoint dell'interfaccia.

Crea un endpoint di interfaccia per Modernizzazione del mainframe AWS

Puoi creare un endpoint di interfaccia per Modernizzazione del mainframe AWS utilizzare la console Amazon VPC o AWS Command Line Interface ().AWS CLI Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per Modernizzazione del mainframe AWS utilizzare il seguente nome di servizio:

```
com.amazonaws.region.m2
```

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API Modernizzazione del mainframe AWS 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 Modernizzazione del mainframe AWS tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito Modernizzazione del mainframe AWS dal tuo VPC, collega una policy endpoint personalizzata all'endpoint di interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (utenti e Account AWS ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni Modernizzazione del mainframe AWS

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Se collegata a un endpoint dell'interfaccia, questa policy concede l'accesso alle operazioni Modernizzazione del mainframe AWS elencate per tutti i principali 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. È possibile raccogliere e tenere traccia dei parametri, 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 CloudTrailacquisisce 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 Guida per l'utente AWS CloudTrail.

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

Metrica	Description
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

Metrica	Description
	Unità: byte al secondo Statistiche valide: Average (Media), Minimum, Maximum

Parametri di applicazioni

Metrica	Description
BatchJobCompletedCount	Il numero di lavori completati durante l'intervallo di tempo. Questa metrica è disponibile per Rocket Software (precedentemente Micro Focus) e per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: somma
BatchJobFailedCount	Il numero di processi non riusciti durante l'intervallo di tempo. Questa metrica è disponibile per Rocket Software e per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: somma

Metrica	Description
JvmMemoryFree	La quantità di memoria disponibile che non è attualmente utilizzata dalla Java Virtual Machine. Questa metrica è disponibile solo per il motore di runtime AWS Transform for mainframe. È disponibile per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: byte Statistiche valide: Average (Media), Minimum, Maximum
JvmMemoryMax	La quantità massima di memoria consentita per la Java Virtual Machine. Questa metrica è disponibile solo per il motore di runtime AWS Transform for mainframe. È disponibile per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: byte Statistiche valide: Average (Media), Minimum, Maximum

Metrica	Description
JvmMemoryUsed	La quantità di memoria utilizzata attivamente dalla Java Virtual Machine. Questa metrica è disponibile solo per il motore di runtime AWS Transform for mainframe. È disponibile per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: byte Statistiche valide: Average (Media), Minimum, Maximum
ProcessesActiveCount	Il numero attivo di processi simultanei di esecuzione del servizio che elaborano le richieste. Questa metrica è disponibile solo per il motore di runtime di Rocket Software. Dimensione: ApplicationID Unità: numero Statistiche valide: somma

Metrica	Description
SessionCount	Il numero di sessioni HTTP per l'applicazione. Questa metrica è disponibile solo per il motore di runtime AWS Transform for mainframe. È disponibile per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: Average (Media), Minimum, Maximum
SharedMemoryFree	La memoria disponibile per il server aziendale per archiviare tutte le informazioni necessarie per eseguire transazioni e processi. Questa metrica è disponibile solo per il motore di runtime di Rocket Software. Dimensione: ApplicationID Unità: Kilobyte Statistiche valide: Average (Media), Minimum, Maximum

Metrica	Description
SharedMemoryTotal	La quantità totale di memoria condivisa allocata al server aziendale per archiviare tutte le informazioni necessarie per eseguire transazioni e processi. Questa metrica è disponibile solo per il motore di runtime di Rocket Software. Dimensione: ApplicationID Unità: Kilobyte Statistiche valide: Average (Media), Minimum, Maximum
ThreadActiveCount	Il numero di thread del motore che elaborano le richieste. Questa metrica è disponibile solo per il motore di runtime AWS Transform for mainframe. È disponibile per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: Average (Media), Minimum, Maximum

Metrica	Description
TransactionCompletedCount	Il numero di transazioni impegnate durante l'intervallo di tempo. Questa metrica è disponibile per Rocket Software e per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: somma
TransactionFailedCount	Il numero di transazioni non riuscite durante l'intervallo di tempo. Questa metrica è disponibile per Rocket Software e per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: numero Statistiche valide: somma

Metrica	Description
TransactionResponseTime	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. Questa metrica è disponibile per Rocket Software e per AWS Transform for mainframe 3.7.0 e versioni successive. Dimensione: ApplicationID Unità: millisecondi Statistiche valide: Average (Media), Minimum, Maximum

Dimensioni

Dimensione	Description
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' AWS API di modernizzazione del mainframe utilizzando AWS CloudTrail

AWS La modernizzazione del mainframe è integrata con AWS CloudTrail un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio in AWS Mainframe Modernization. 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 CloudTrail utente.

AWS Informazioni 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. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail 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 della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail 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 AWS Mainframe 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 AWS servizio.

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'`CreateApplication`azione.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAII6WZTHGYAEXAMPLE",
        "arn": "arn:aws:sts::444455556666:assumed-role/Admin/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAII6WZTHGYAEXAMPLE",
                "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-jkl2m3456789",
        "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/
lsfhmwhw7fffrosff2lncwqcua",
        "applicationId": "lsfhmwhw7fffrosff2lncwqcua"
    },
    "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 nella AWS modernizzazione del mainframe

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 AWS Transform for mainframe e Rocket Software.

Argomenti

- Errore di risoluzione dei problemi: timeout in attesa dello sblocco del nome del set di dati
- Errore di risoluzione dei problemi: impossibile accedere all'URL di un'applicazione
- Risoluzione dei problemi: AWS Transform for mainframe refactor non si apre dalla console
- Errore di risoluzione dei problemi: ambiente non integro
- Risoluzione dei problemi di licenza per Rocket Software (precedentemente Micro Focus)

Errore di risoluzione dei problemi: timeout in attesa dello sblocco del nome del set di dati

Questa pagina descrive come risolvere l'errore quando un'altra applicazione in un ambiente blocca un set di dati condiviso.

- Motore: AWS Transform per mainframe
- Componente: Blusam

Se vedi questo errore nei CloudWatch log di Amazon per un'applicazione di modernizzazione del AWS mainframe che utilizza il motore AWS Transform for mainframe 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 arresta in modo anomalo o fallisce in altro modo e non rilascia il blocco.

Cerca un'applicazione fallita e controlla se utilizza lo stesso set di dati indicato nel messaggio di errore. Verificate 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.

Cause comuni

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 `example-app-2` l'annullamento del tentativo di scrittura e l'insorgenza di un'eccezione di timeout. Questa situazione di stallo impedisce a tutte le applicazioni di arrivare. `example-dataset-1`

Risoluzione

Per risolvere immediatamente la situazione, puoi forzare lo sblocco. Per evitare che si verifichi una situazione simile in futuro, è possibile configurare due parametri che controllano il meccanismo di riparazione Blusam automatica.

Forza il rilascio della serratura

Il gestore dei Blusam blocchi utilizza Amazon ElastiCache (Redis OSS) 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 ElastiCache nella ElastiCache console all'indirizzo 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.

Comando	Scopo
CHIAVI * *YOUR_DATASET_NAME*	Ottieni una chiave di blocco del set di dati.
DEL *THE_RETURNED_KEY*	Eliminare un blocco del set di dati.
FLUSHDB	Pulisci l'intero Redis. **Warning** 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

Configura il Blusam meccanismo di riparazione automatica

Il gestore dei Blusam blocchi 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 gestione dei Blusam blocchi per il controllo dei blocchi. Tutti i Blusam blocchi in ElastiCache entrata hanno una data e un'ora di 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 Transform for mainframe, vedere. AWS Esempio di definizione dell'applicazione Transform for mainframe

Blusamgestore delle serrature

Nel contesto di un ambiente di runtime di modernizzazione del AWS mainframe che utilizza il modello High Availability, un'applicazione AWS Transform for mainframe potrebbe essere implementata più volte. Per le applicazioni che gestiscono set di Blusam dati, potrebbero verificarsi problemi di accesso simultaneo. Il gestore dei Blusam blocchi 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 che possiede il 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 Blusam dei blocchi gestisce l'accesso da più thread in una determinata applicazione utilizzando lo stesso meccanismo di blocco.

Errore di risoluzione dei problemi: impossibile accedere all'URL di un'applicazione

Questa pagina descrive come risolvere l'errore quando non riesci ad accedere all'URL di un'applicazione di modernizzazione del AWS mainframe in esecuzione.

- Motore: AWS Transform per mainframe e Rocket Software (precedentemente 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.

Cause comuni

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.

Risoluzione

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/
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:

Tipo

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 sta chiamando l'applicazione. Puoi scegliere MyIP dal menu a discesa. Se hai ancora problemi di timeout, prova a scegliere Anywhere o Anywhere IPV4. 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.

Risoluzione dei problemi: AWS Transform for mainframe refactor non si apre dalla console

Questa pagina descrive come risolvere la pagina di refactor di AWS Transform for mainframe che non si apre dalla console di modernizzazione del AWS mainframe.

- Motore: Transform per mainframe AWS
- Componente: AWS Transform for mainframe refactor

Quando si tenta di accedere a AWS Transform for mainframe refactor dalla console di modernizzazione del AWS mainframe, non si apre e la nuova scheda viene chiusa immediatamente.

Cause comuni

Il ruolo che stai utilizzando per accedere a AWS Transform for mainframe refactor non dispone di autorizzazioni sufficienti.

Risoluzione

Allega una policy IAM al ruolo per consentirgli di accedere a AWS Transform for mainframe refactor. Assicurati che la policy includa almeno le seguenti autorizzazioni.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "m2:GetSignedBluinsightsUrl"
            ],
            "Resource": "*"
            }
    ]
}
```

Assicurati di sostituire `region` e `account` con la corretta Regione AWS e Account AWS.

Errore di risoluzione dei problemi: ambiente non integro

Questa pagina descrive come risolvere l'errore quando si riceve una notifica che indica che uno degli ambienti di modernizzazione del AWS mainframe non è integro.

- Motore: AWS Transform per mainframe e Rocket Software (precedentemente Micro Focus)
- Componente: ambienti

Se ricevete una notifica che indica che uno dei vostri ambienti di modernizzazione del AWS mainframe non è più integro, questo vale per voi. Riceverai una notifica tramite una di queste fonti:

- Lo stato dell'ambiente non integro viene visualizzato nella console di modernizzazione del AWS mainframe.
- Notifica via e-mail sullo stato dell'ambiente non integro da. AWS Health

- Puoi vedere un evento correlato di AWS Mainframe Modernization nella tua AWS Health dashboard, sotto Lo stato del tuo account.

Cause comuni

L'errore si verifica quando le risorse del tuo AWS account associate all'ambiente di modernizzazione del AWS mainframe sono inaccessibili. Un motivo comune di questo problema è che le risorse relative all'ambiente vengono modificate o eliminate.

Risoluzione

Per indicazioni specifiche, utilizza il codice di errore fornito nell'e-mail inviata da AWS Health o tramite la console di modernizzazione del AWS mainframe.

Codice di errore:

- Archiviazione irraggiungibile

Questo errore indica che lo storage collegato (Amazon Elastic File System o Amazon FSx file system) per l'ambiente non è stato montato correttamente. Per verificare i dettagli sull'ambiente non integro, completa i seguenti passaggi:

1. Apri la console di modernizzazione del AWS mainframe all'indirizzo. https://console.aws.amazon.com/m2/
2. Seleziona l'ambiente non integro e scegli Configurazione.
3. Scegli Attached Storage per visualizzare le risorse di archiviazione associate a questo ambiente.
4. Controlla le configurazioni relative alla rete, come il gruppo di sicurezza, la sottorete e Amazon VPC associati allo storage. Se queste configurazioni non sono corrette, prova a ripristinarle per risolvere il problema.

> **Note**
>
> Se lo storage è stato eliminato, l'ambiente non può essere ripristinato. In questo caso, è consigliabile prendere in considerazione l'eliminazione dell'ambiente non integro.

Risoluzione dei problemi di licenza per Rocket Software (precedentemente Micro Focus)

Questa pagina descrive come risolvere i problemi di licenza con il motore Rocket Software Runtime

- Motore: Rocket Software
- Componente: Amazon EC2

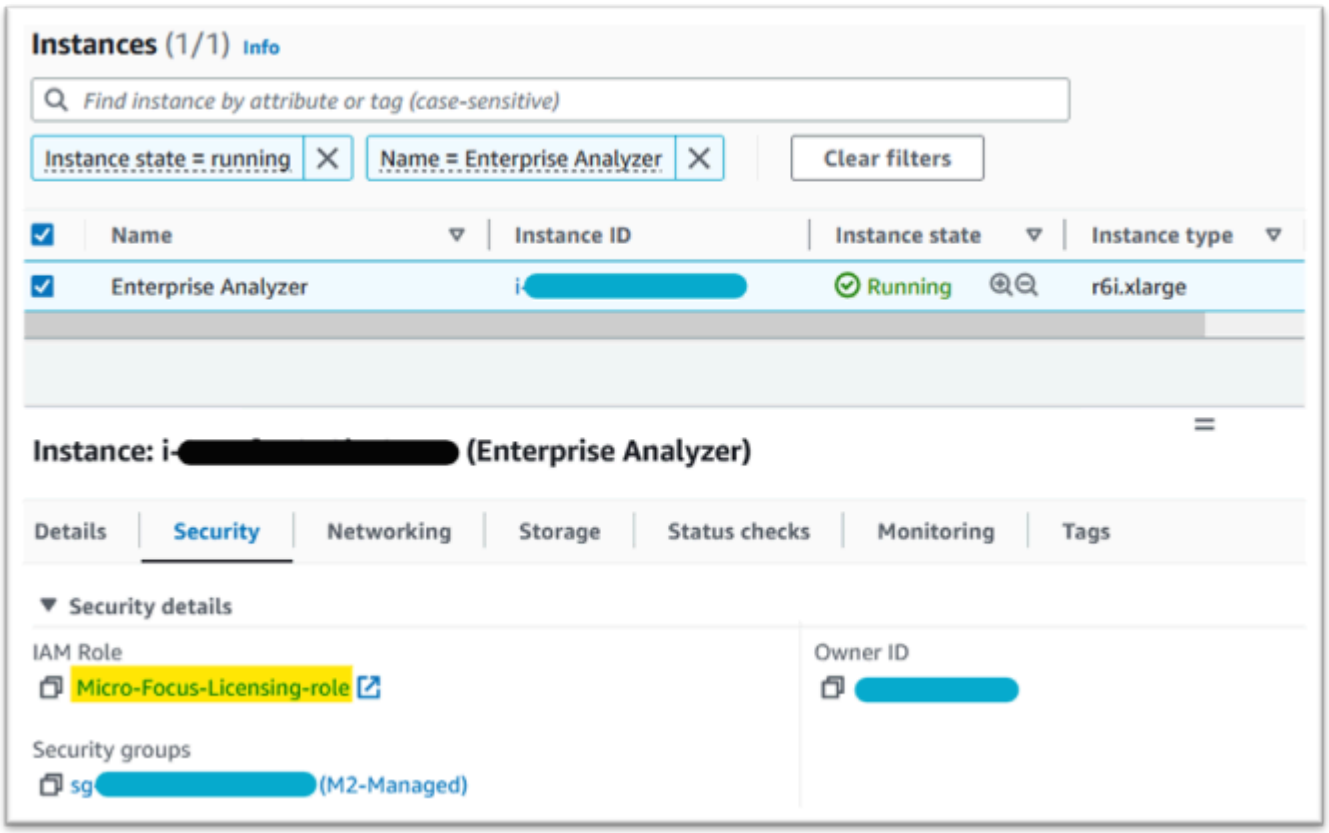
In caso di problemi di accesso o utilizzo di AMIs, le seguenti informazioni potrebbero esserti utili.

Argomenti

- Verifica che l'istanza Amazon EC2 abbia il ruolo di licenza IAM
- Usa l'analizzatore di raggiungibilità
- Esegui il license-daemon
- Problemi di licenza con Enterprise Server o Enterprise Build Tools su Linux dopo l'applicazione di patch al sistema operativo

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.



Usa l'analizzatore di raggiungibilità

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 SLF4 J 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 SLF4 J e cerca la prima eccezione.

Su Linux esegui:

```
java -jar /var/microfocuslicensing/bin/aws-license-daemon.jar
```

Ignora i messaggi SLF4 J 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.

Problemi di licenza con Enterprise Server o Enterprise Build Tools su Linux dopo l'applicazione di patch al sistema operativo

In caso di problemi di licenza con Enterprise Server o Enterprise Build Tools su Linux dopo l'applicazione di patch al sistema operativo, aggiorna il daemon della licenza scaricando ed eseguendo uno script di patch. A tale scopo, utilizzate i seguenti comandi nel prompt dei comandi:

```
sudo curl https://d148y999krizvm.cloudfront.net/patch/v8/linux/patch.sh -o /var/
microfocuslicensing/bin/patch.sh
sudo chmod +x /var/microfocuslicensing/bin/patch.sh
sudo /var/microfocuslicensing/bin/patch.sh
sudo ./startmfcesd.sh
```

> **Note**
>
> Questo script di patch funzionerà anche con la versione 9 anche se il percorso di download è per la versione 8.

Cronologia dei documenti per la guida per l' AWS utente alla modernizzazione del mainframe

La tabella seguente descrive le versioni della documentazione per la modernizzazione del AWS mainframe.

Modifica	Descrizione	Data
AWS La modernizzazione del mainframe non sarà più aperta a nuovi clienti a partire dal 7 novembre 2025.	AWS La modernizzazione del mainframe non sarà più aperta a nuovi clienti a partire dal 7 novembre 2025. Se desideri utilizzare AWS Mainframe Modernization Service, registrati prima di tale data. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta Modifica della disponibilità di AWS Mainframe Modernization.	7 ottobre 2025
AWS Note di rilascio 4.7.0 e 4.8.0 di Transform for mainframe	La versione 4.7.0 di AWS Transform for mainframe Runtime e Transform for mainframe AWS Transformation Engines include aggiornamenti delle dipendenze di base a Spring Boot 3.4.2 e Angular 19, che migliorano la sicurezza e le prestazioni e modernizzano al contempo il framework applicativo frontend. La versione 4.8.0 include	18 maggio 2025

	funzionalità chiave come il supporto per la trasformazione del database IBM IMS e una nuova funzionalità di cache in memoria che migliora le prestazioni per i programmi di modernizzazione a uso intensivo di dati. AS400	
AWS Note di rilascio 4.6.0 di Transform for mainframe	Questa versione del motore di AWS trasformazione Transform for mainframe Runtime e AWS Transform for mainframe si concentra su nuove funzionalità e miglioram enti per zOS e. AS400	24 gennaio 2025
AWS Note di rilascio di Transform for mainframe 4.5.0	Questa versione del motore di AWS trasformazione Transform for mainframe Runtime e AWS Transform for mainframe si concentra sulle funzionalità chiave del supporto JCL, sulle directory di associazione e sul supporto dei gruppi di attivazione per le applicazioni modernizzate AS/400 e sulle dipendenze aggiornate.	20 dicembre 2024
AWS Domande frequenti su Transform for mainframe	Scopri la funzionalità di refactoring di AWS Transform for mainframe con questo elenco completo di. FAQs	20 dicembre 2024

Amazon Q Developer Transform per mainframe	Puoi scoprire la funzionalità Transform for mainframe di Amazon Q Developer che ti consente di modernizzare le tue applicazioni mainframe COBOL precedenti in applicazioni Java.	2 dicembre 2024
AWS Note di rilascio di Transform for mainframe 4.4.0	Questa versione di AWS Transform for mainframe Runtime and Transformation Engine si concentra sull'aggiornamento delle dipendenze critiche e delle tecnologie supportate, migliorando al contempo le prestazioni in molteplici funzionalità.	13 novembre 2024
AWS Implementa Transform for mainframe Runtime su container	Puoi imparare a configurare AWS Transform for mainframe Runtime sui container per distribuirlo su Amazon ECS (gestito da Amazon EC2 o) AWS Fargate e Amazon EKS gestito da Amazon EC2.	28 ottobre 2024
Credenziali utente LDAP aggiornate	Ora puoi creare e gestire le credenziali utente LDAP per l'autenticazione e l'autorizzazione utilizzando la AWS console o AWS CLI (o l'SDK).	21 ottobre 2024

Configura l'applicazione gestita da Rocket Software	Ora puoi configurare le tue applicazioni con il motore di runtime Rocket Software per personalizzare proprietà aggiuntive, incluse le integrazioni.	21 ottobre 2024
Ripiattaforma con il runtime NTT DATA Unix	Scopri come usare Amazon Machine Images (AMIs) per creare un ambiente personalizzato per il rehosting e il replatforming delle applicazioni mainframe AWS utilizzando NTT DATA.	25 settembre 2024
AWS Test delle applicazioni di modernizzazione del mainframe (IAM)	Puoi scoprire come gestire l'accesso per i test delle applicazioni di modernizzazione del AWS mainframe con le funzionalità e le policy IAM disponibili.	20 settembre 2024
AWS Transform per versioni mainframe	Questa sezione della guida per l'utente di AWS Mainframe Modernization contiene tutti i dettagli sulle versioni di AWS Transform for mainframe , le note di rilascio di AWS Transform for mainframe , le istruzioni per l'aggiornamento di AWS Transform for mainframe e il ciclo di vita complessivo di Transform for mainframe. AWS	16 settembre 2024

AWS Ciclo di vita dei componenti di modernizzazione del mainframe	Questa pagina illustra il ciclo di vita di ogni componente di modernizzazione del AWS mainframe, inclusi gli aggiornamenti di versione, il piano di rilascio generale e i piani di fine supporto e ritiro.	5 settembre 2024
Conversione di Assembler con mLogica	AWS La conversione del codice di modernizzazione del mainframe con MLogica è una funzionalità di modernizzazione del AWS mainframe che converte automaticamente il codice di Assembler del mainframe in COBOL. z/OS	22 luglio 2024
Versione Application Testing GA	Documenti sulla disponibilità generale per i test delle applicazioni. AWS Mainframe Modernization Application Testing fornisce test automatizzati di equivalenza funzionale e per i progetti di migrazione. Questa versione include la pagina di protezione dei dati, i flussi di lavoro della console e gli aggiornamenti ad altre pagine del documento dopo l'anteprima.	12 giugno 2024

Managed Runtime for Rocket Softwaretutorial aggiornato	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 Rocket Software.	5 febbraio 2024
Note di rilascio per AWS Transform for mainframe Runtime and Modernization Tools versione 3.9.0.	Questa versione di AWS Transform for mainframe 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
Trasferisci file tra mainframe e AWS	È stata rilasciata una nuova funzionalità per il trasferimento di file dal mainframe di origine a. AWS	27 novembre 2023
Gestisci le transazioni per le applicazioni	È stata rilasciata una nuova funzionalità per visualizzare e modificare le transazioni per le applicazioni per la modernizzazione del AWS mainframe.	16 ottobre 2023

Note di rilascio per AWS Transform for mainframe Runtime and Modernization Tools versione 3.6.0.	Questa versione di AWS Transform for mainframe Runtime and Modernization Tools offre nuove funzionalità sia per gli zOS che per le migrazioni AS400 precedenti, 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
È ora possibile distribuire una nuova versione di un'applicazione quando l'applicazione viene interrotta.	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
AWS Transform for mainframe runtime confezionato per una distribuzione più semplice di Amazon EC2	AWS La modernizzazione del mainframe con AWS Transform for mainframe runtime è ora disponibile con maggiore flessibilità per la configurazione dello stack completo e la distribuzione sulle istanze Amazon EC2 del tuo. Account AWS	06 luglio 2023

Single Sign-On per Transform for mainframe Transform for mainframe refactor. AWSAWS	AWS Transform for mainframe AWS Transform for mainframe Refactor è disponibile tramite Single Sign-On. Console di gestione AWS	31 marzo 2023
Versione GA	Versione GA della Mainframe Modernization User Guide AWS .	8 giugno 2022
Versione iniziale	Versione iniziale (anteprim a pubblica) della AWS Mainframe Modernization User Guide.	30 novembre 2021