



Benutzerhandbuch

# AWS Mainframe-Modernisierung



# AWS Mainframe-Modernisierung: Benutzerhandbuch

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

Die Handelsmarken und Handelsaufmachung von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, durch die Kunden irregeführt werden könnten oder Amazon in schlechtem Licht dargestellt oder diskreditiert werden könnte. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

---

# Table of Contents

Was ist AWS Mainframe-Modernisierung? .....	1
Merkmale der Mainframe-Modernisierung AWS .....	2
Muster .....	3
Wie fange ich mit AWS der Mainframe-Modernisierung an .....	3
Zugehörige Services .....	4
Zugriff auf AWS Mainframe-Modernisierung .....	5
Verwenden Sie Mainframe Modernization zum ersten Mal? AWS .....	5
Preisgestaltung .....	5
Einrichtung der AWS Mainframe-Modernisierung .....	6
Melden Sie sich an für ein AWS-Konto .....	6
Erstellen Sie einen Benutzer mit Administratorzugriff .....	6
Erste Schritte .....	9
Tutorial: Verwaltete Laufzeit für AWS Blu Age .....	9
Voraussetzungen .....	10
Schritt 1: Laden Sie die Demo-Anwendung hoch .....	10
Schritt 2: Erstellen Sie die Anwendungsdefinition .....	10
Schritt 3: Erstellen Sie eine Laufzeitumgebung .....	11
Schritt 4: Erstellen Sie eine Anwendung .....	16
Schritt 5: Stellen Sie eine Anwendung bereit .....	18
Schritt 6: Starten Sie eine Anwendung .....	21
Schritt 7: Greifen Sie auf die Anwendung zu .....	21
Schritt 8: Testen der Anwendung .....	22
Bereinigen von -Ressourcen .....	24
Tutorial: Verwaltete Laufzeit für Micro Focus .....	24
Voraussetzungen .....	25
Schritt 1: Einen Amazon S3 S3-Bucket erstellen und laden .....	25
Schritt 2: Eine Datenbank erstellen und konfigurieren .....	27
Schritt 3: Erstellen und konfigurieren Sie ein AWS KMS key .....	29
Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel .....	30
Schritt 5: Erstellen Sie eine Laufzeitumgebung .....	31
Schritt 6: Erstellen Sie eine Anwendung .....	38
Schritt 7: Stellen Sie eine Anwendung bereit .....	44
Schritt 8: Datensätze importieren .....	46

Schritt 9: Starten Sie eine Anwendung .....	52
Schritt 10: Connect zur CardDemo CICS-Anwendung her .....	53
Bereinigen von -Ressourcen .....	61
Nächste Schritte .....	62
Modernisierungsansatz .....	63
Phase bewerten .....	63
Phase der Mobilisierung .....	64
Phase der Migration und Modernisierung .....	64
Phase „Betrieb und Optimierung“ .....	65
Konzepte .....	66
Anwendung .....	66
Definition der Anwendung .....	66
Batch-Auftrag .....	67
Konfiguration .....	68
Dataset .....	68
Umgebung .....	68
Mainframe-Modernisierung .....	68
Reise zur Migration .....	68
Befestigungspunkt .....	68
Automatisiertes Refactoring .....	69
Plattformwechsel .....	69
Ressource .....	69
Runtime-Engine .....	69
AWS Blu Age Refactoring .....	70
AWS Versionshinweise zu Blu Age .....	71
Versionshinweise 4.1.0 .....	72
Runtime-Version 4.1.0 .....	73
Version 4.1.0 der Modernisierungstools .....	78
Versionshinweise 4.0.0 .....	80
Runtime-Version 4.0.0 .....	81
Version 4.0.0 der Modernisierungstools .....	87
Versionshinweise 3.10.0 .....	90
Runtime-Version 3.10.0 .....	91
Version 3.10.0 der Modernisierungstools .....	94
Versionshinweise 3.9.0 .....	95
Runtime-Version 3.9.0 .....	96

Version 3.9.0 der Modernisierungstools .....	101
Versionshinweise 3.8.0 .....	104
Runtime-Version 3.8.0 .....	105
Version 3.8.0 der Modernisierungstools .....	108
Versionshinweise 3.7.0 .....	110
Runtime-Version 3.7.0 .....	111
Version 3.7.0 der Modernisierungstools .....	113
Versionshinweise 3.6.0 .....	116
Runtime-Version 3.6.0 .....	116
Version 3.6.0 der Modernisierungstools .....	120
Versionshinweise 3.5.0 .....	122
Runtime-Version 3.5.0 .....	123
Version 3.5.0 der Modernisierungstools .....	126
AWS Anweisungen zum Upgrade auf Blu Age .....	129
Migration von 3.10.0 auf 4.0.0 .....	129
AWS Blu Age Runtime-Konzepte .....	131
Architektur auf hohem Niveau .....	131
Modernisierte Anwendungsstruktur .....	136
Datenvereinfacher .....	173
AWS Konfiguration von Blu Age Runtime .....	181
Grundlagen der Anwendungskonfiguration .....	182
Vorrang der Anwendung .....	184
JNDI für Datenbanken .....	184
AWS Geheimnisse verwenden .....	185
Andere Dateien (Groovy, SQL usw.) .....	190
Zusätzliche Webanwendung .....	191
Eigenschaften aktivieren .....	192
Konfiguration der Sicherheit für Gapwalk-Anwendungen .....	247
AWS Blu-Age-Laufzeit-APIs .....	264
URLs erstellen .....	264
Gapwalk-Anwendung .....	265
REST-Endpunkte der Blusam Application Console .....	285
JICS-Anwendungskonsole .....	307
Datenstrukturen .....	328
AWS Einrichtung von Blu Age Runtime (nicht verwaltet) .....	338
AWS Voraussetzungen für die Laufzeit von Blu Age .....	338

AWS Einführung in Blu Age Runtime .....	339
Anforderungen an die Einrichtung der Infrastruktur .....	344
Bereitstellung auf Amazon ECS, verwaltet von AWS Fargate .....	351
Bereitstellung auf Amazon EC2 .....	359
Testen Sie die Anwendung PlanetsDemo .....	374
Ändern Sie den Quellcode mit Blu Age Developer IDE .....	377
Tutorial: AppStream 2.0 für AWS Blu Age Developer IDE einrichten .....	378
Tutorial: Verwenden Sie AWS Blu Age Developer auf AppStream 2.0 .....	384
Replatforming von Micro Focus .....	401
Einrichtung von Micro Focus Runtime (auf Amazon EC2) .....	401
Voraussetzungen .....	402
Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3 .....	402
Fordere die Aktualisierung der Zulassungsliste für das Konto an .....	405
Die AWS Identity and Access Management Rolle wird erstellt .....	406
Gewähren Sie License Manager die erforderlichen Berechtigungen .....	413
Abonnieren Sie die Amazon Machine Images .....	414
Starten Sie eine Micro Focus-Instance für die AWS Mainframe-Modernisierung .....	417
Subnetz oder VPC ohne Internetzugang .....	424
Behebung von Lizenzproblemen .....	430
Tutorial: Richten Sie AppStream 2.0 für Enterprise Analyzer und Enterprise Developer ein .....	432
Voraussetzungen .....	434
Schritt 1: Holen Sie sich die AppStream 2.0-Bilder .....	434
Schritt 2: Erstellen Sie den Stack mithilfe der Vorlage AWS CloudFormation .....	434
Schritt 3: Erstellen Sie einen Benutzer in AppStream 2.0 .....	438
Schritt 4: Melden Sie sich bei AppStream 2.0 an .....	439
Schritt 5: Buckets in Amazon S3 verifizieren (optional) .....	441
Nächste Schritte .....	442
Bereinigen von -Ressourcen .....	442
Richten Sie Enterprise Analyzer ein .....	442
Inhalt des Bilds .....	443
Voraussetzungen .....	444
Schritt 1: Einrichtung .....	445
Schritt 2: Erstellen Sie den Amazon S3 S3-basierten virtuellen Ordner unter Windows .....	445
Schritt 3: Erstellen Sie eine ODBC-Quelle für die Amazon RDS-Instance .....	447
Nachfolgende Sitzungen .....	449
Problembehandlung bei der Workspace-Verbindung .....	449

Bereinigen von -Ressourcen .....	454
Richten Sie Enterprise Developer ein .....	454
Inhalt des Bilds .....	455
Voraussetzungen .....	456
Schritt 1: Einrichtung durch einzelne Enterprise Developer-Benutzer .....	456
Schritt 2: Erstellen Sie den Amazon S3-basierten virtuellen Ordner unter Windows (optional) .....	457
Schritt 3: Klonen Sie das Repository .....	458
Nachfolgende Sitzungen .....	459
Bereinigen von -Ressourcen .....	460
Richten Sie die AppStream 2.0-Automatisierung ein .....	460
Richten Sie die Automatisierung beim Sitzungsstart ein .....	460
Richten Sie die Automatisierung am Ende der Sitzung ein .....	461
Datensätze als Tabellen anzeigen .....	461
Voraussetzungen .....	462
Schritt 1: ODBC-Verbindung zum Micro Focus-Datenspeicher (Amazon RDS-Datenbank) einrichten .....	462
Schritt 2: Erstellen Sie die Datei MFDBFH.cfg .....	465
Schritt 3: Erstellen Sie eine Strukturdatei (STR) für Ihr Copybook-Layout .....	465
Schritt 4: Erstellen Sie eine Datenbankansicht mithilfe der Strukturdatei (STR) .....	468
Schritt 5: Micro Focus-Datensätze als Tabellen und Spalten anzeigen .....	468
Verwenden Sie Vorlagen mit Enterprise Developer .....	469
Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält .....	470
Anwendungsfall 2 — Verwendung der COBOL-Projektvorlage ohne Quellkomponenten .....	472
Anwendungsfall 3 — Verwendung des vordefinierten COBOL-Projekts, das auf die Quellordner verweist .....	474
Verwenden der JSON-Vorlage für die Regionsdefinition .....	477
Tutorial: Richten Sie den Build für das BankDemo Beispiel ein .....	480
Voraussetzungen .....	482
Schritt 1: Amazon S3 S3-Buckets erstellen .....	482
Schritt 2: Erstellen Sie die Build-Spezifikationsdatei .....	483
Schritt 3: Laden Sie die Quelldateien hoch .....	484
Schritt 4: IAM-Richtlinien erstellen .....	484
Schritt 5: Erstellen Sie eine IAM-Rolle .....	487
Schritt 6: Hängen Sie die IAM-Richtlinien an die IAM-Rolle an .....	488

Schritt 7: Erstellen Sie das CodeBuild Projekt .....	488
Schritt 8: Starten Sie den Build .....	489
Schritt 9: Laden Sie die Ausgabeartefakte herunter .....	490
Bereinigen von -Ressourcen .....	490
Tutorial: Einrichtung einer CI/CD-Pipeline zur Verwendung mit Micro Focus Enterprise	
Developer .....	491
Voraussetzungen .....	492
Erstellen Sie eine grundlegende Infrastruktur für die CI/CD-Pipeline .....	494
Erstellen Sie ein AWS CodeCommit Repository und eine CI/CD-Pipeline .....	497
Erstellung AppStream von Enterprise Developer 2.0 .....	502
Einrichtung und Test für Enterprise Developer .....	502
Übung 1: Verbessern Sie die Kreditberechnung in der BANKDEMO-Anwendung .....	508
Übung 2: Extrahieren Sie die Kreditberechnung in der Anwendung BankDemo .....	512
Bereinigen von -Ressourcen .....	516
Batch-Dienstprogramme .....	517
Binärer Standort .....	518
M2SFTP-Batch-Hilfsprogramm .....	518
M2WAIT Batch-Hilfsprogramm .....	525
TXT2PDF Batch-Hilfsprogramm .....	527
M2DFUTIL Batch-Hilfsprogramm .....	533
M2RUNCMD Batch-Hilfsprogramm .....	540
Datenreplikation mit Precisely .....	545
Voraussetzungen .....	545
Abonnieren Sie das Amazon Machine Image .....	545
Starten Sie die AWS Mainframe Modernization Datenreplikation mit Precisely .....	546
Eine IAM-Richtlinie erstellen .....	547
Erstellen einer IAM-Rolle .....	548
Hängen Sie die IAM-Rolle an die Amazon EC2 EC2-Instance an .....	548
Integration von Charon .....	550
Einführung in Charon-SSP .....	550
Unterstützte Gastbetriebssysteme .....	552
Voraussetzungen für die Charon-SSP Cloud-Instanz .....	553
Voraussetzungen für die Instanz .....	554
Erstellen und Konfigurieren einer AWS Cloud-Instanz für Charon (neue GUI) .....	556
Allgemeine Voraussetzungen .....	556
Verwenden von AWS Management Console , um eine neue Instanz zu starten .....	557



Umgestaltung der Plattform mit NTT DATA .....	563
Voraussetzungen .....	563
Abonnieren Sie das Amazon Machine Image .....	563
Starten Sie die AWS Mainframe Modernization Replatform mit der NTT DATA-Instanz .....	564
Erste Schritte mit NTT Data .....	565
Anwendungen .....	567
Erstellen einer Anwendung .....	568
Erstellen einer Anwendung .....	568
Stellen Sie eine Anwendung bereit .....	569
Stellen Sie eine Anwendung bereit .....	569
Aktualisieren einer Anwendung .....	571
Aktualisieren einer Anwendung .....	571
Löschen Sie eine Anwendung aus einer Umgebung .....	572
Löschen Sie eine Anwendung aus einer Umgebung .....	572
Löschen einer Anwendung .....	573
Löschen einer Anwendung .....	573
Reichen Sie Batch-Jobs für Bewerbungen ein .....	573
Reichen Sie einen Batch-Job ein .....	574
Starten Sie einen Batch-Job neu .....	575
Importieren Sie Datensätze für Anwendungen .....	576
Importieren Sie einen Datensatz .....	576
Transaktionen für Anwendungen verwalten .....	577
Transaktionen für Anwendungen verwalten .....	577
AWS Ressourcen für eine migrierte Anwendung erstellen .....	579
Erforderliche Berechtigungen .....	579
Amazon-S3-Bucket .....	579
Datenbank .....	580
AWS Key Management Service Schlüssel .....	581
AWS Secrets Manager geheim .....	581
Die verwaltete Anwendung konfigurieren .....	582
Struktur der von AWS Blu Age verwalteten Anwendungen .....	582
Konfiguration des Zugriffs auf Dienstprogramme für verwaltete Anwendungen .....	584
Konfigurieren Sie zusätzliche Eigenschaften .....	593
Referenz zur Anwendungsdefinition .....	615
Allgemeiner Kopfbereich .....	615
Überblick über den Definitionsbereich .....	617

AWS Beispiel für eine Blu-Age-Anwendungsdefinition .....	617
AWS Einzelheiten zur Blu Age-Definition .....	618
Micro Focus-Anwendungsdefinition .....	624
Einzelheiten zur Micro Focus-Definition .....	625
Referenz zur Definition von Datensätzen .....	633
Gemeinsame Eigenschaften .....	634
Beispiel für ein Datensatz-Anforderungsformat für VSAM .....	636
Beispiel für ein Datensatz-Anforderungsformat für GDG Base .....	638
Beispiel für ein Anforderungsformat für Datensätze für PS- oder GDG-Generationen .....	639
Beispiel für ein Datensatz-Anforderungsformat für PO .....	641
Verwaltete Laufzeitumgebungen .....	643
Erstellen Sie eine Laufzeitumgebung .....	644
Erstellen Sie eine Laufzeitumgebung .....	644
Aktualisieren Sie eine Laufzeitumgebung .....	647
Aktualisieren Sie eine Laufzeitumgebung .....	647
Wartungsfenster .....	648
Stoppen Sie eine Laufzeitumgebung .....	649
Beenden Sie eine Laufzeitumgebung .....	649
Starten Sie eine Laufzeitumgebung neu .....	651
Starten Sie eine Laufzeitumgebung neu .....	651
Löschen Sie eine Laufzeitumgebung .....	652
Löschen Sie eine Laufzeitumgebung .....	652
Testen von Anwendungen .....	653
Was sind Anwendungstests? .....	653
Verwenden Sie Application Testing zum ersten Mal? .....	654
Vorteile von Application Testing .....	654
Integration mit AWS CloudFormation .....	655
Wie funktioniert das Testen von Anwendungen .....	656
Zugehörige Services .....	4
Zugriff auf Anwendungstests .....	657
Preise für Anwendungstests .....	658
Konzepte für das Testen von Anwendungen .....	658
Testfall .....	659
Testszenario .....	659
Projekt testen .....	660
Ursprünglicher Zustand .....	660

Aufzeichnen (Erfassen) .....	660
Erneut abspielen .....	661
Compare .....	661
Datenbankvergleiche .....	661
Vergleiche von Datensätzen .....	662
Status des Vergleichs .....	662
Regeln für die Gleichwertigkeit .....	663
Vergleich von Datensätzen im Endzustand .....	663
Vergleiche zwischen Datenbanken zum Stand und zum Fortschritt .....	663
Funktionale Äquivalenz (FE) .....	664
Online-Bildschirmvergleiche von 3270 .....	664
Aufnahme läuft .....	664
Daten erneut abspielen .....	664
Referenzdaten .....	664
Aufnahmen, Wiedergeben und Vergleichen .....	665
Unterschiede .....	666
Äquivalenzen .....	666
Quellanwendung .....	666
Zielanwendung .....	666
Tutorial: Richten Sie ein CardDemo .....	666
Voraussetzungen .....	667
Schritt 1: Bereiten Sie die Einrichtung vor CardDemo .....	667
Schritt 2: Erstellen Sie alle erforderlichen Ressourcen .....	668
Schritt 3: Stellen Sie die Anwendung bereit und starten Sie sie .....	669
Schritt 4: Erste Daten importieren .....	669
Schritt 5: Connect zur CardDemo Anwendung her .....	670
Tutorial: Wiederholung und Vergleich auf Blu Age mit AWS CardDemo .....	671
Schritt 1: Besorgen Sie sich AWS Blu Age Amazon EC2 Amazon Machine Image (AMI) .....	672
Schritt 2: Starten Sie eine Amazon EC2 EC2-Instance mit dem AWS Blu Age AMI .....	672
Schritt 3: Laden Sie CardDemo abhängige Dateien auf S3 hoch .....	674
Schritt 4: Datenbanken laden und die Anwendung initialisieren CardDemo .....	674
Schritt 5: Starten Sie AWS Blu Age Runtime CloudFormation .....	676
Schritt 6: Testen der AWS Blu Age Amazon EC2 EC2-Instance .....	679
Schritt 7: Stellen Sie sicher, dass die vorherigen Schritte korrekt abgeschlossen wurden .....	680
Schritt 8. Erstellen Sie eine Ausgangsbedingung .....	681
Schritt 9: Erstellen Sie den Testfall .....	681

Schritt 10: Erstellen Sie ein Testszenario .....	682
Schritt 11: Nehmen Sie Ihr Testszenario auf .....	682
Schritt 12: Wiedergeben und vergleichen .....	683
Unterstützte Codepages für Datensätze .....	683
Übertragung von Dateien .....	695
Was ist Dateiübertragung? .....	695
Vorteile von AWS Mainframe Modernization File Transfer .....	695
So funktioniert AWS Mainframe Modernization File Transfer .....	696
Installieren Sie einen File Transfer Agent .....	697
Schritt 1: Melden Sie sich bei ISPF an .....	697
Schritt 2: Ordnen Sie dem z/FS einen Datensatz zu .....	697
Schritt 3: Formatieren Sie den Datensatz als z/FS .....	698
Schritt 4: Definieren Sie das Dateisystem auf z/OS .....	698
Schritt 5: Mounten Sie das Dateisystem .....	698
Schritt 6: Überprüfen Sie den Mount .....	699
Schritt 7: Geben Sie OMVs ein .....	699
Schritt 8: Legen Sie die Umgebungsvariable für das Installationsverzeichnis des Agenten fest .....	699
Schritt 9: Legen Sie die Umgebungsvariable für das Arbeitsverzeichnis fest .....	699
Schritt 10: Erstellen Sie das Arbeitsverzeichnis .....	699
Schritt 11: Kopieren Sie das Tar-Paket AWS Mainframe Modernization in das Arbeitsverzeichnis auf z/OS .....	700
Schritt 12: Gehen Sie vom Root-Benutzer aus .....	700
Konfigurieren Sie Berechtigungen und STC .....	701
Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten .....	702
Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll .....	702
Konfiguration des Agenten .....	704
Endpunkte für die Datenübertragung .....	706
Erstellen Sie einen Datenübertragungsendpunkt .....	706
Aufgaben übertragen .....	708
Übertragungsaufgaben erstellen .....	709
Übertragungsaufgaben anzeigen .....	711
Tutorial: Erste Schritte mit File Transfer .....	712
Übersicht .....	712
Schritt 1: Übertragen Sie das Tar-Paket der Agent-Binärdateien von AWS auf die logische Mainframe-Partition .....	713

Schritt 2: Konfigurieren Sie den File Transfer Agent auf dem Quell-Mainframe .....	713
Schritt 3: Erstellen Sie einen Datenübertragungsendpunkt .....	713
Schritt 4: Erstellen Sie eine Übertragungsaufgabe .....	713
Schritt 5: Status der Übertragungsaufgabe einsehen .....	713
Unterstützte Quell- und Zielcodepages .....	714
Mainframe-Datensatztypen .....	714
Unterstützte Codepages .....	714
Sicherheit .....	716
Datenschutz .....	717
Daten, die AWS Mainframe Modernization sammelt .....	718
Datenverschlüsselung im Ruhezustand für den AWS Mainframe-Modernisierungsservice ...	719
So verwendet AWS Mainframe Modernization Zuschüsse in AWS KMS .....	722
Einen kundenverwalteten Schlüssel erstellen .....	724
Angabe eines vom Kunden verwalteten Schlüssels für die AWS Mainframe- Modernisierung .....	725
AWS Verschlüsselungskontext für die Mainframe-Modernisierung .....	726
Überwachen Ihrer Verschlüsselungsschlüssel .....	728
Weitere Informationen .....	744
Verschlüsselung während der Übertragung .....	744
Identitäts- und Zugriffsverwaltung .....	744
Zielgruppe .....	745
Authentifizierung mit Identitäten .....	746
Verwalten des Zugriffs mit Richtlinien .....	750
So funktioniert die AWS Mainframe-Modernisierung mit IAM .....	753
Beispiele für identitätsbasierte Richtlinien .....	767
Fehlerbehebung .....	771
Verwenden von serviceverknüpften Rollen .....	772
Compliance-Validierung .....	776
Ausfallsicherheit .....	777
Sicherheit der Infrastruktur .....	777
AWS PrivateLink .....	778
Überlegungen .....	779
Erstellen eines Schnittstellenendpunkts .....	779
Erstellen einer Endpunktrichtlinie .....	779
Überwachen .....	781
Überwachung mit CloudWatch .....	781

Metriken zur Laufzeitumgebung .....	782
Anwendungsmetriken .....	783
Dimensionen .....	788
CloudTrail protokolliert .....	788
AWS Informationen zur Mainframe-Modernisierung finden Sie unter CloudTrail .....	788
Grundlegendes zu Einträgen in der Protokolldatei zur AWS Mainframe-Modernisierung .....	789
Fehlerbehebung .....	792
Fehler: Beim Warten auf die Entsperrung des Datensatznamens ist eine Zeitüberschreitung aufgetreten .....	792
Wie tritt dieser Fehler auf .....	792
Woher wissen Sie, ob dies Ihre Situation ist? .....	793
Was können Sie tun? .....	793
Erzwingen Sie das Lösen des Schlosses .....	793
Konfigurieren Sie den auto Reparaturmechanismus von Blusam .....	794
Blusam Locks Manager .....	795
Auf die URL einer Anwendung kann nicht zugegriffen werden .....	796
Wie tritt dieser Fehler auf .....	796
Woher wissen Sie, ob dies Ihre Situation ist? .....	796
Was können Sie tun? .....	796
AWS Blu Insights wird nicht von der Konsole aus geöffnet .....	797
Wie tritt dieser Fehler auf .....	798
Was können Sie tun? .....	798
Dokumentverlauf .....	799
.....	dcccii

# Was ist AWS Mainframe-Modernisierung?

AWS Mainframe Modernization hilft Ihnen dabei, Ihre Mainframe-Anwendungen zu verwalteten Laufzeitumgebungen zu modernisieren. AWS Er bietet Tools und Ressourcen, die Sie bei der Planung und Implementierung von Migration und Modernisierung unterstützen. Sie können Ihre vorhandenen Mainframe-Anwendungen analysieren, sie mithilfe von COBOL oder PL/I entwickeln oder aktualisieren und eine automatisierte Pipeline für die kontinuierliche Integration und kontinuierliche Bereitstellung (CI/CD) der Anwendungen implementieren. Sie können je nach den Bedürfnissen Ihrer Kunden zwischen automatisiertem Refactoring und Replatforming-Mustern wählen. Wenn Sie als Berater einem Kunden bei der Migration seiner Mainframe-Workloads helfen, können Sie Mainframe-Modernisierungstools für alle Phasen der Migration und Modernisierung einsetzen AWS , von der ersten Planung bis hin zum Cloud-Betrieb nach der Migration.

Mithilfe der AWS Mainframe-Modernisierung können Sie die Laufzeitumgebung AWS für Ihre Mainframe-Anwendungen effizient erstellen und verwalten sowie Ihre modernisierten Anwendungen verwalten und überwachen.

## Themen

- [Merkmale der Mainframe-Modernisierung AWS](#)
- [Muster](#)
- [Wie fange ich mit AWS der Mainframe-Modernisierung an](#)
- [Zugehörige Services](#)
- [Zugriff auf AWS Mainframe-Modernisierung](#)
- [Verwenden Sie Mainframe Modernization zum ersten Mal? AWS](#)
- [Preisgestaltung](#)

### Note

Haben Sie sich für Ihr AWS Mainframe-Modernisierungsprojekt an Partner oder AWS Professional Services für die Mainframe-Migration gewandt? Falls nicht, empfehlen wir Ihnen dringend, Experten für Ihr Projekt zu beauftragen.

- [AWS Kompetenzpartner für Mainframe-Modernisierung](#)
- [AWS Professional Services](#)

Die Funktionen und Anwendungsfälle der AWS Mainframe-Modernisierung unterstützen einen evolutionären Modernisierungsansatz, der kurzfristige Erfolge durch verbesserte Agilität und zahlreiche Möglichkeiten zur späteren Optimierung und Innovation bietet. Weitere Informationen finden Sie unter [Modernisierungsansatz](#).

## Merkmale der Mainframe-Modernisierung AWS

AWS Die Funktionen der Mainframe-Modernisierung unterstützen die folgenden Anwendungsfälle:

- **Bewertung:** Die Bewertungsfunktionen für die AWS Mainframe-Modernisierung können Sie bei der Bewertung, dem Umfang und der Planung eines Migrations- und Modernisierungsprojekts unterstützen.
- **Refactor:** Mit Hilfe von AWS Blu Age können Sie Refactoring verwenden, um ältere Anwendungsprogrammiersprachen zu konvertieren, Makroservices oder Microservices zu erstellen und Benutzeroberflächen (UIs) und Anwendungssoftware-Stacks zu modernisieren.

AWS Blu Insights ist jetzt über Single Sign-On verfügbar. AWS Management Console Sie müssen keine separaten AWS Blu Insights-Anmeldeinformationen mehr verwalten. Sie können sowohl auf die Funktionen AWS AWS Blu Age Codebase als auch auf das AWS Management Console Transformation Center direkt über zugreifen.

- **Replatform:** Mithilfe der Micro Focus Enterprise-Lösung können Sie die Anwendung so portieren, dass ein Großteil des Anwendungsquellcodes ohne Änderungen neu kompiliert wird.
- **Entwickler-IDE:** AWS Mainframe Modernization bietet eine integrierte Entwicklungsumgebung (IDE) auf Abruf, sodass Entwickler durch intelligentes Bearbeiten und Debuggen, sofortige Codekompilierung und Komponententests schneller Code schreiben können.
- **Verwaltete Laufzeit:** Die verwaltete Laufzeitumgebung von AWS Mainframe Modernization überwacht Ihre Cluster kontinuierlich, um die Workloads des Unternehmens mit selbstreparierender Rechenleistung und automatisierter Skalierung am Laufen zu halten.
- **Continuous Integration and Delivery (CI/CD):** Die CI/CD-Funktion von AWS Mainframe Modernization unterstützt Anwendungsentwicklungsteams dabei, Codeänderungen häufiger und zuverlässiger durchzuführen. Dadurch wird die Migrationsgeschwindigkeit beschleunigt, die Qualität erhöht und der Aufwand für die Veröffentlichung neuer Geschäftsfunktionen reduziert. time-to-market
- **Integrationen mit anderen AWS Services:** AWS Mainframe Modernization unterstützt AWS CloudFormation, AWS PrivateLink und sorgt AWS Key Management Service für wiederholbare Bereitstellungen sowie mehr Sicherheit und Compliance.



- **Erweiterte Verfügbarkeit:** Die AWS Mainframe-Modernisierung ist jetzt in den Ländern USA Ost (Ohio), USA West (Nordkalifornien), Asien-Pazifik (Mumbai), Asien-Pazifik (Seoul), Asien-Pazifik (Singapur), Asien-Pazifik (Tokio), Europa (London) und Europa (Paris) verfügbar.

Weitere Informationen zu den Funktionen der AWS Mainframe-Modernisierung finden Sie unter.

<https://aws.amazon.com/mainframe-modernization/features/>

## Muster

Das automatisierte Refactoring-Muster, das von AWS Blu Age unterstützt wird, konzentriert sich auf die Beschleunigung der Modernisierung, indem der gesamte ältere Anwendungsstapel und seine Datenschicht in eine moderne Java-basierte Anwendung umgewandelt werden, wobei die funktionale Äquivalenz gewahrt bleibt. Während dieser automatisierten Transformation wird eine mehrstufige Anwendung mit einem Angular-basierten Frontend, einem API-fähigen Java-Backend und einer Datenschicht erstellt, die auf moderne Datenspeicher zugreift. Der Refactoring-Prozess bietet die gleiche Funktionalität wie der ältere Stack, um die Projektautomatisierung zu erhöhen, was zu Geschwindigkeit, Qualität und geringeren Kosten führt, sodass Geschäftsvorteile schneller erzielt werden können. Weitere Informationen finden Sie unter [AWS Mainframe Modernization Automated Refactor](#).

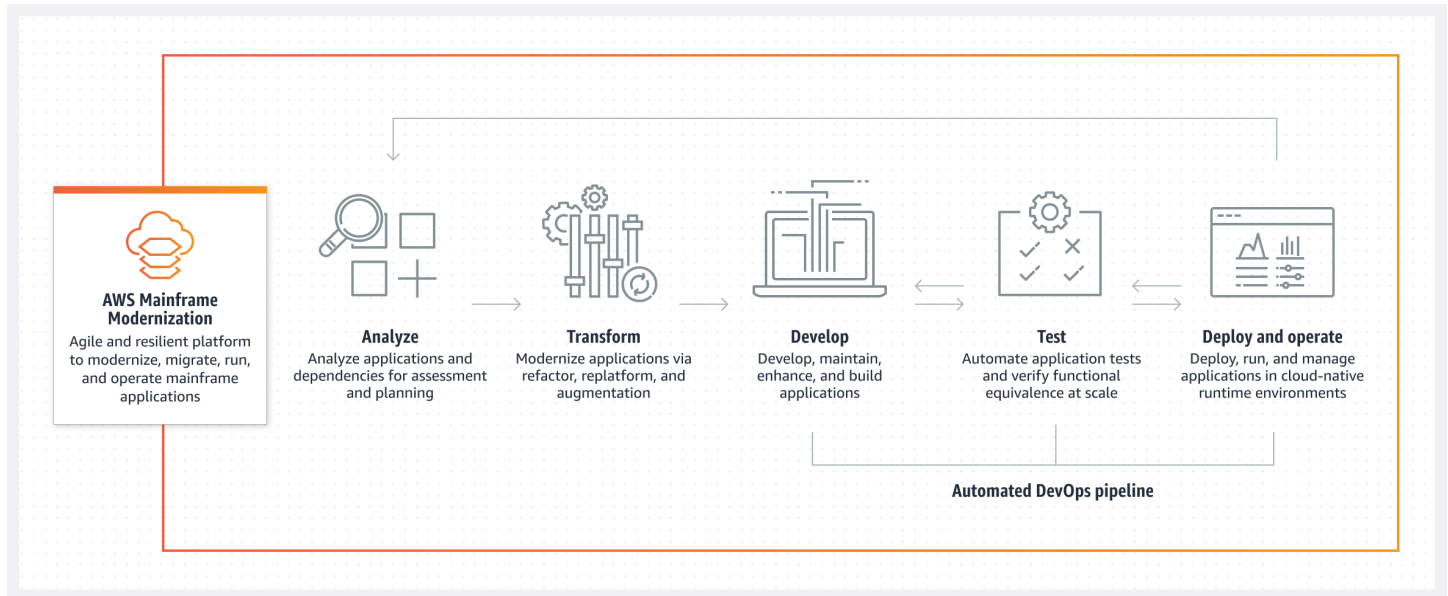
Das Replatforming-Pattern, das auf der Micro Focus Enterprise Suite basiert, konzentriert sich auf die Beibehaltung der Anwendungssprache, des Codes und der Artefakte, um die Auswirkungen auf die Anwendungsressourcen und -teams so gering wie möglich zu halten. Es hilft Kunden, ihr Anwendungswissen und ihre Fähigkeiten aufrechtzuerhalten. Die Anwendungsänderungen sind zwar begrenzt, aber dieses Muster erleichtert auch die Modernisierung der Infrastruktur und der Prozesse. Die Infrastruktur wird auf einen modernen, cloudbasierten Managed Service umgestellt und gleichzeitig werden die Prozesse so geändert, dass sie den bewährten Verfahren für die Anwendungsentwicklung und den IT-Betrieb folgen. Weitere Informationen finden Sie unter [AWS Mainframe Modernization Replatform](#).

## Wie fange ich mit AWS der Mainframe-Modernisierung an

Versuchen Sie es! Wir bieten Tutorials und Beispielanwendungen, damit Sie sich ein Bild davon machen können, was die AWS Mainframe-Modernisierung bietet. Wählen Sie entweder das [Tutorial: Verwaltete Laufzeit für AWS Blu Age](#) oder das, [Tutorial: Verwaltete Laufzeit für Micro Focus](#) um ein vollständiges step-by-step Tutorial zu erhalten.

Wenn Sie an automatisiertem Refactoring interessiert sind, schauen Sie sich die AWS Blu Age-Tools unter an. [BluInsights](#) Sie können AppStream 2.0 auch für den Zugriff auf die AWS Blu Age Developer IDE oder die Tools Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten.

Die Tutorials und Beispielanwendungen geben Ihnen nur einen Eindruck davon, was die AWS Mainframe-Modernisierung bietet. Wenn Sie bereit sind, ein Modernisierungsprojekt zu starten, finden Sie [Modernisierungsansatz](#) weitere Informationen zu den Phasen und Aufgaben eines Modernisierungsprojekts unter.



## Zugehörige Services

Zusätzlich zu Blu Insights für automatisiertes Refactoring können Sie die folgenden AWS Services mit AWS Mainframe Modernization nutzen.

- Amazon RDS zum Hosten Ihrer migrierten Datenbanken.
- Amazon S3 zum Speichern von Anwendungsbinärdateien und Definitionsdateien.
- Amazon FSx oder Amazon EFS zum Speichern von Anwendungsdaten.
- Amazon AppStream für den Zugriff auf die Tools Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer.
- AWS CloudFormation für die automatisierte DevOps Pipeline, mit der Sie CI/CD für Ihre migrierten Anwendungen einrichten können.
- AWS Migration Hub

- [AWS DMS für die Migration Ihrer Datenbanken.](#)

## Zugriff auf AWS Mainframe-Modernisierung

[Derzeit können Sie über die Konsole unter AWS https://console.aws.amazon.com/m2/ auf Mainframe-Modernisierung zugreifen.](https://console.aws.amazon.com/m2/) Eine Liste der Regionen, in denen AWS Mainframe-Modernisierung verfügbar ist, finden Sie unter [Endpunkte und Kontingente für die AWS Mainframe-Modernisierung](#) in der [Allgemeine Amazon Web Services-Referenz](#)

## Verwenden Sie Mainframe Modernization zum ersten Mal? AWS

Wenn Sie AWS Mainframe Modernization zum ersten Mal verwenden, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- [Erste Schritte](#)
- [Einrichtung](#)

## Preisgestaltung

AWS Mainframe Modernization berechnet Gebühren für die Nutzung von Instanzen, die die verwalteten Laufzeitumgebungen unterstützen. Darüber hinaus bietet AWS Mainframe Modernization einige Tools ohne zusätzliche Kosten. Sie sind für Gebühren verantwortlich, die für andere AWS Dienste anfallen, die Sie im Zusammenhang mit der AWS Mainframe-Modernisierung nutzen. AWS wird Sie 30 Tage im Voraus informieren, bevor Preisänderungen für die Nutzung von AWS Mainframe Modernization in Kraft treten. Weitere Informationen finden Sie unter [Mainframe-Modernisierung](#) mit AWS

Mit AWS Blu Insights zahlen Sie für die Nutzung des Transformation Centers. Weitere Informationen finden Sie unter [Preise für die AWS Mainframe-Modernisierung](#).

# Einrichtung der AWS Mainframe-Modernisierung

Bevor Sie mit der Nutzung von AWS Mainframe Modernization beginnen können, müssen Sie oder Ihr Administrator einige Schritte ausführen.

## Themen

- [Melden Sie sich an für ein AWS-Konto](#)
- [Erstellen Sie einen Benutzer mit Administratorzugriff](#)

## Melden Sie sich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie <https://portal.aws.amazon.com/billing/signup>.
2. Folgen Sie den Online-Anweisungen.

Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Tasten eingeben.

Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontos wird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Aus Sicherheitsgründen sollten Sie einem Benutzer Administratorzugriff zuweisen und nur den Root-Benutzer verwenden, um [Aufgaben auszuführen, für die Root-Benutzerzugriff erforderlich](#) ist.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Sie können jederzeit Ihre aktuelle Kontoaktivität anzeigen und Ihr Konto verwalten. Rufen Sie dazu <https://aws.amazon.com/> auf und klicken Sie auf Mein Konto.

## Erstellen Sie einen Benutzer mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

## Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich [AWS Management Console](#) als Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter [Anmelden als Root-Benutzer](#) im AWS-Anmeldung Benutzerhandbuch zu.

2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für den Root-Benutzer.

Anweisungen finden Sie unter [Aktivieren eines virtuellen MFA-Geräts für Ihren AWS-Konto Root-Benutzer \(Konsole\)](#) im IAM-Benutzerhandbuch.

## Erstellen Sie einen Benutzer mit Administratorzugriff

1. Aktivieren Sie das IAM Identity Center.

Anweisungen finden Sie unter [Aktivieren AWS IAM Identity Center](#) im AWS IAM Identity Center Benutzerhandbuch.

2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden [Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis](#) im AWS IAM Identity Center Benutzerhandbuch.

## Melden Sie sich als Benutzer mit Administratorzugriff an

- Um sich mit Ihrem IAM-Identity-Center-Benutzer anzumelden, verwenden Sie die Anmelde-URL, die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM-Identity-Center-Benutzer erstellt haben.

Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie [im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal](#).

## Weisen Sie weiteren Benutzern Zugriff zu

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

Anweisungen finden Sie im Benutzerhandbuch unter [Einen Berechtigungssatz erstellen](#).AWS IAM Identity Center

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

Anweisungen finden [Sie im AWS IAM Identity Center Benutzerhandbuch unter Gruppen hinzufügen](#).

# Erste Schritte mit der AWS Mainframe-Modernisierung

Um mit der AWS Mainframe-Modernisierung zu beginnen, können Sie Tutorials folgen, die Sie mit dem Service und den einzelnen Runtime-Engines vertraut machen.

## Themen

- [Tutorial: Verwaltete Laufzeit für AWS Blu Age](#)
- [Tutorial: Verwaltete Laufzeit für Micro Focus](#)

Weitere Informationen finden Sie in den folgenden Tutorials.

- [Tutorial: Den Micro Focus-Build für die BankDemo Beispielanwendung einrichten](#)
- [Tutorial: Einrichtung einer CI/CD-Pipeline zur Verwendung mit Micro Focus Enterprise Developer](#)

## Tutorial: Verwaltete Laufzeit für AWS Blu Age

Dieses Tutorial zeigt, wie Sie eine modernisierte AWS Blu Age-Anwendung in einer AWS Mainframe-Modernisierungs-Laufzeitumgebung bereitstellen.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Laden Sie die Demo-Anwendung hoch](#)
- [Schritt 2: Erstellen Sie die Anwendungsdefinition](#)
- [Schritt 3: Erstellen Sie eine Laufzeitumgebung](#)
- [Schritt 4: Erstellen Sie eine Anwendung](#)
- [Schritt 5: Stellen Sie eine Anwendung bereit](#)
- [Schritt 6: Starten Sie eine Anwendung](#)
- [Schritt 7: Greifen Sie auf die Anwendung zu](#)
- [Schritt 8: Testen der Anwendung](#)
- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

[Um dieses Tutorial abzuschließen, laden Sie das Demo-Anwendungsarchiv PlanetsDemo -v1.zip herunter.](#)

Für den Zugriff auf die laufende Demo-Anwendung ist ein moderner Browser erforderlich. Ob Sie diesen Browser von Ihrem Desktop oder von einer Amazon Elastic Compute Cloud-Instance aus ausführen, beispielsweise innerhalb der VPC, bestimmt Ihre Sicherheitseinstellungen.

## Schritt 1: Laden Sie die Demo-Anwendung hoch

Laden Sie die Demo-Anwendung in einen Amazon S3 S3-Bucket hoch. Stellen Sie sicher, dass sich dieser Bucket in demselben Bucket befindet, in AWS-Region dem Sie die Anwendung bereitstellen werden. Das folgende Beispiel zeigt einen Bucket namens planetsdemo mit einem key prefix oder Ordner namens v1 und einem Archiv mit dem Namen. p1anetsdemo-v1.zip

Amazon S3 > Buckets > planetsdemo > v1/

v1/ Copy S3 URI

Objects | Properties

**Objects (1)**  
Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Refresh Copy S3 URI Copy URL Download Open Delete Actions Create folder Upload

Find objects by prefix

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

### Note

Der Ordner im Bucket ist erforderlich.

## Schritt 2: Erstellen Sie die Anwendungsdefinition

Um eine Anwendung für die verwaltete Laufzeit bereitzustellen, benötigen Sie eine Anwendungsdefinition für die AWS Mainframe-Modernisierung. Diese Definition ist eine JSON-Datei,



die den Speicherort und die Einstellungen der Anwendung beschreibt. Das folgende Beispiel ist eine solche Anwendungsdefinition für die Demo-Anwendung:

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

Ändern Sie den s3-bucket Eintrag in den Namen des Buckets, in dem Sie die ZIP-Datei der Beispielanwendung gespeichert haben.

Weitere Informationen zur Anwendungsdefinition finden Sie unter [AWS Beispiel für eine Blu-Age-Anwendungsdefinition](#).

### Schritt 3: Erstellen Sie eine Laufzeitumgebung

Gehen Sie wie folgt vor, um die Laufzeitumgebung für die AWS Mainframe-Modernisierung zu erstellen:

1. Öffnen Sie die [AWS Mainframe Modernization Console](#).
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Umgebung erstellen möchten. Dies AWS-Region muss mit der Region übereinstimmen, in der Sie den S3-Bucket erstellt haben. [Schritt 1: Laden Sie die Demo-Anwendung hoch](#)
3. Wählen Sie unter Mainframe-Anwendungen modernisieren die Option Refactor with Blu Age und dann Erste Schritte aus.

## Modernize mainframe applications

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

Choose an option to get started.






- Refactor with Blu Age
- Replatform with Micro Focus

**Get started**

4. Wählen Sie unter Wie kann AWS Mainframe Modernization helfen? die Option Laufzeitumgebung bereitstellen und erstellen aus.

### How can AWS Mainframe Modernization help?

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

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

#### Deploy [Info](#)

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

5. Wählen Sie in der linken Navigationsleiste Umgebungen und anschließend Umgebung erstellen aus. Geben Sie auf der Seite Basisinformationen angeben einen Namen und eine Beschreibung

für Ihre Umgebung ein, und stellen Sie dann sicher, dass die AWS Blu Age-Engine ausgewählt ist. Optional können Sie der erstellten Ressource Tags hinzufügen. Wählen Sie anschließend Weiter.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
**Specify basic information**

Step 2  
Specify configurations

Step 3 - Optional  
Attach storage

Step 4  
Review and create

## Specify basic information [Info](#)

### Name and description

Environment name

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


Environment description - optional

The description can be up to 500 characters.


### Engine options

Select Engine

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



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



AWS Blu Age Version

6. Wählen Sie auf der Seite „Konfigurationen angeben“ die Option Standalone-Laufzeitumgebung aus.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
**Specify configurations**

Step 3 - Optional  
Attach storage

Step 4  
Review and create

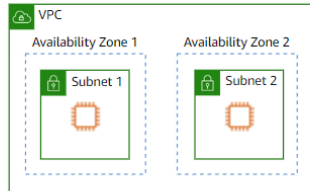
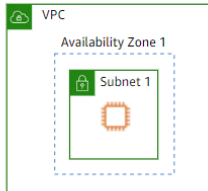
## Specify configurations [Info](#)

### Availability [Info](#)

Choose the availability pattern for your environment.

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

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



7. Nehmen Sie unter Sicherheit und Netzwerk die folgenden Änderungen vor:

- Wählen Sie In dieser Umgebung bereitgestellte Anwendungen öffentlich zugänglich machen aus. Diese Option weist der Anwendung eine öffentliche IP-Adresse zu, sodass Sie von Ihrem Desktop aus darauf zugreifen können.
- Wählen Sie eine VPC aus. Sie können die Standardeinstellung verwenden.
- Wählen Sie zwei Subnetze aus. Stellen Sie sicher, dass die Subnetze die Zuweisung von öffentlichen IP-Adressen zulassen.
- Wählen Sie eine Sicherheitsgruppe aus. Sie können die Standardeinstellung verwenden. Stellen Sie sicher, dass die von Ihnen gewählte Sicherheitsgruppe den Zugriff von der Browser-IP-Adresse auf den Port ermöglicht, den Sie in der `listener` Eigenschaft der Anwendungsdefinition angegeben haben. Weitere Informationen finden Sie unter [Schritt 2: Erstellen Sie die Anwendungsdefinition](#).

## Security and network

Allow applications deployed to this environment to be publicly accessible.

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

Default vpc-

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

Choose subnets

subnet- X

subnet- X

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

Choose security groups

default X  
default VPC security group

Wenn Sie von außerhalb der von Ihnen ausgewählten VPC auf die Anwendung zugreifen möchten, stellen Sie sicher, dass die Regeln für eingehende Zugriffe für diese VPC ordnungsgemäß konfiguriert sind. Weitere Informationen finden Sie unter [Auf die URL einer Anwendung kann nicht zugegriffen werden](#).

8. Wählen Sie Weiter.
9. Behalten Sie unter Speicher anhängen — Optional die Standardauswahl bei und wählen Sie Weiter.

AWS Mainframe Modernization > Environments > Create Environment

Step 1  
Specify basic information

Step 2  
Specify configurations

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

Step 4  
Review and create

## Attach storage - *Optional* Info

### EFS storage

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

No EFS associated with this environment.

You can add up to 1 more EFS.

### FSx storage

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

No EFS associated with this environment.

You can add up to 1 more FSx.

10. Wählen Sie unter **Wartung planen** die Option **Keine Präferenz** und dann **Weiter** aus.
11. Überprüfen Sie unter **Überprüfen und erstellen** die Informationen und wählen Sie dann **Umgebung erstellen** aus.

## Schritt 4: Erstellen Sie eine Anwendung

1. Navigieren Sie zu **AWS Mainframe Modernization** in der **AWS Management Console**.
2. Wählen Sie im Navigationsbereich **Applications (Anwendungen)** und anschließend **Create a new application (Neue Anwendung erstellen)**. Geben Sie auf der Seite **Basisinformationen** angeben einen Namen und eine Beschreibung für die Anwendung ein und stellen Sie sicher, dass die **AWS Blu Age-Engine** ausgewählt ist. Wählen Sie anschließend **Weiter**.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name


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

Application description - *optional*


The maximum length is 500 characters.

### Engine type

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



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



3. Kopieren Sie auf der Seite „Ressourcen und Konfigurationen angeben“ die aktualisierte JSON-Anwendungsdefinition, in [the section called “Schritt 2: Erstellen Sie die Anwendungsdefinition”](#) der Sie sie erstellt haben, und fügen Sie sie ein.

AWS Mainframe Modernization > Applications > Create application

Step 1  
Specify basic information

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

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

### Resources and configurations

Choose an approach to define the application

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

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

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

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

- Überprüfen Sie unter Überprüfen und erstellen Ihre Auswahl und wählen Sie dann Anwendung erstellen aus.

## Schritt 5: Stellen Sie eine Anwendung bereit

Nachdem Sie sowohl die Laufzeitumgebung als auch die Anwendung für die AWS Mainframe-Modernisierung erfolgreich erstellt haben und beide den Status Verfügbar haben, können Sie die Anwendung in der Laufzeitumgebung bereitstellen. Führen Sie dazu die folgenden Schritte aus:

- Navigieren Sie in der AWS Management Console zu AWS Mainframe Modernization. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus. Die Seite mit der Liste der Umgebungen wird angezeigt.



AWS Mainframe Modernization > Environments

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

< 1 > ⚙️

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

2. Wählen Sie die zuvor erstellte Laufzeitumgebung aus. Die Seite mit den Umgebungsdetails wird angezeigt.

3. Wählen Sie Anwendung bereitstellen aus.

AWS Mainframe Modernization > Environments > planets-demo-env

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

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

**Environment** [Info](#)

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

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

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

4. Wählen Sie die zuvor erstellte Anwendung und anschließend die Version aus, in der Sie Ihre Anwendung bereitstellen möchten. Wählen Sie dann Deploy (Bereitstellen) aus.

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

## Deploy application Info

You have selected the following application:

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

### Available versions (0)

Choose a version from the list.

< 1 > 

Version

Creation time

No versions  
No versions to display

### Environments (1) Info

< 1 > 

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

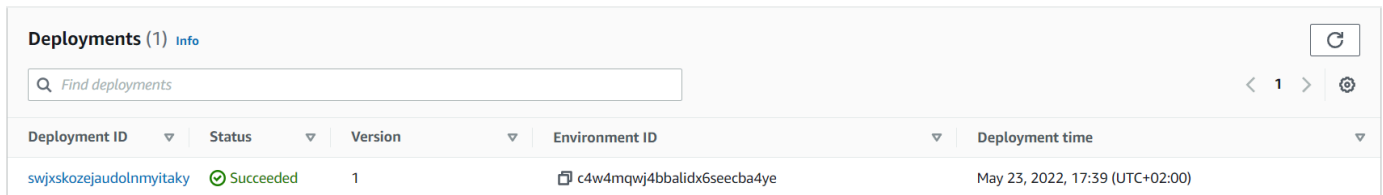
Cancel

Deploy

5. Warten Sie, bis die Anwendung ihre Bereitstellung abgeschlossen hat. Sie sehen ein Banner mit der Meldung Die Anwendung wurde erfolgreich bereitgestellt.

## Schritt 6: Starten Sie eine Anwendung

1. Navigieren Sie im zu AWS Mainframe Modernization AWS Management Console und wählen Sie Applications aus.
2. Wählen Sie Ihre Anwendung aus und gehen Sie dann zu Bereitstellungen. Der Status der Anwendung sollte „Erfolgreich“ lauten.



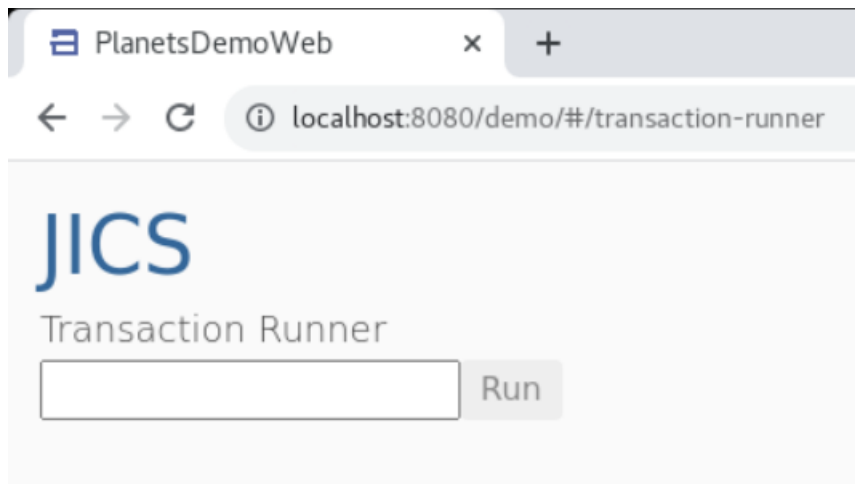
Deployment ID	Status	Version	Environment ID	Deployment time
swjxskozejaudolnmyitaky	Succeeded	1	c4w4mqwj4bbalidx6seecba4ye	May 23, 2022, 17:39 (UTC+02:00)

3. Wählen Sie Aktionen und anschließend Anwendung starten.

## Schritt 7: Greifen Sie auf die Anwendung zu

1. Warten Sie, bis sich die Anwendung im Status Running befindet. Sie sehen ein Banner mit der Meldung Die Anwendung wurde erfolgreich gestartet.
2. Kopieren Sie den DNS-Hostnamen der Anwendung. Sie finden diesen Hostnamen im Abschnitt Anwendungsinformationen der Anwendung.
3. Navigieren Sie in einem Browser zu `http://{hostname}:{portname}/PlanetsDemo-web-1.0.0/`, wo:
  - `hostname` ist der zuvor kopierte DNS-Hostname.
  - `portname` ist der Tomcat-Port, der in der Anwendungsdefinition definiert ist, in der Sie sie erstellt haben. [Schritt 2: Erstellen Sie die Anwendungsdefinition](#)

Der JICS-Bildschirm wird angezeigt.



Wenn Sie nicht auf die Anwendung zugreifen können, finden Sie weitere Informationen unter [Auf die URL einer Anwendung kann nicht zugegriffen werden](#).

#### Note

Wenn auf die Anwendung nicht zugegriffen werden kann und in der Regel für eingehende Nachrichten für die Sicherheitsgruppe auf Port 8196 die Option „Meine IP“ ausgewählt ist, geben Sie die Regel an, um Datenverkehr von LB I/P auf Port 8196 zuzulassen.

## Schritt 8: Testen der Anwendung

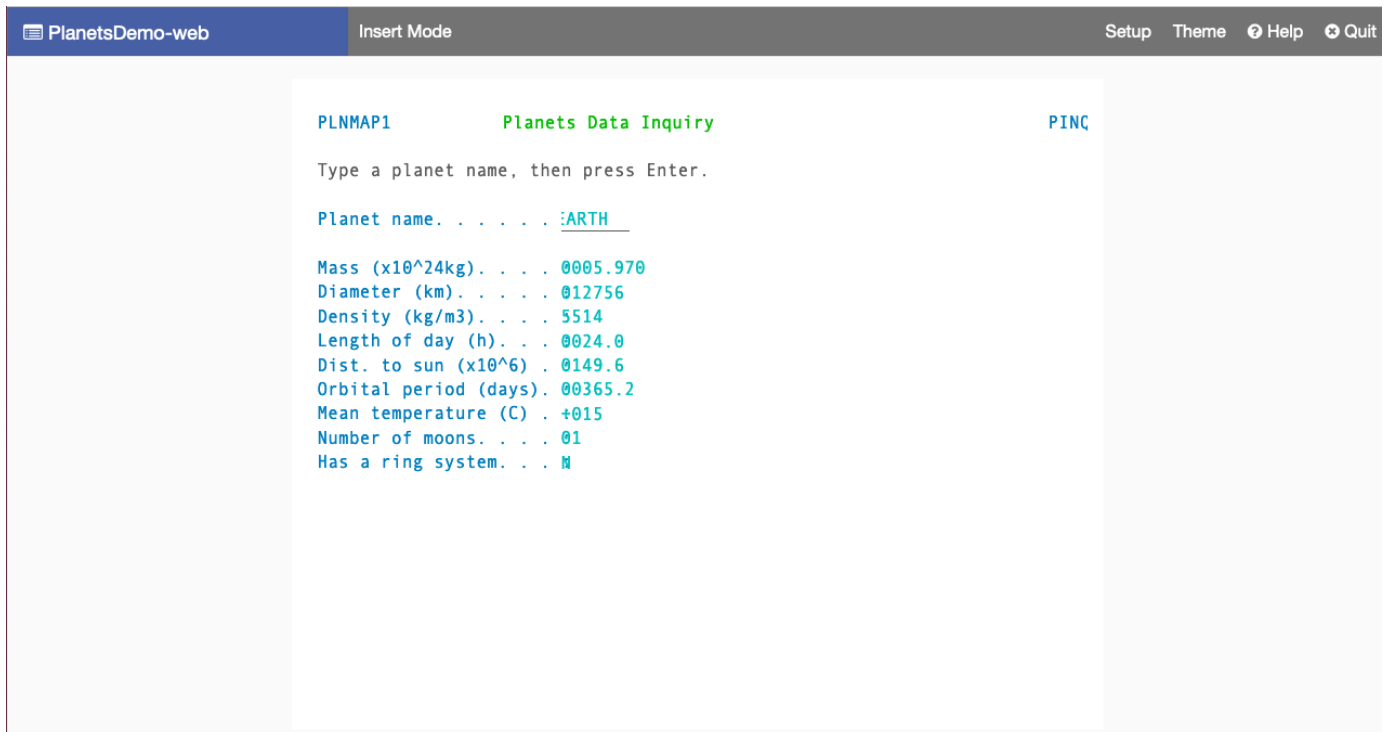
In diesem Schritt führen Sie eine Transaktion in der migrierten Anwendung aus.

1. Geben Sie auf dem JICS-Bildschirm den Text PINQ in das Eingabefeld ein und wählen Sie Ausführen (oder drücken Sie die Eingabetaste), um die Anwendungstransaktion zu starten.

Der Bildschirm der Demo-App sollte erscheinen.



2. Geben Sie einen Planetennamen in das entsprechende Feld ein und drücken Sie die Eingabetaste.



Sie sollten Details über den Planeten sehen.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, um zusätzliche Kosten zu vermeiden. Führen Sie dazu die folgenden Schritte aus:

- Wenn die AWS Mainframe-Modernisierungsanwendung noch läuft, beenden Sie sie.
- Löschen Sie die -Anwendung. Weitere Informationen finden Sie unter [Löschen Sie eine AWS Mainframe-Modernisierungsanwendung](#).
- Löschen Sie die Laufzeitumgebung. Weitere Informationen finden Sie unter [Löschen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#).

## Tutorial: Verwaltete Laufzeit für Micro Focus

Dieses Tutorial zeigt, wie Sie die CardDemo Beispielanwendung in einer von AWS Mainframe Modernization verwalteten Laufzeitumgebung mit der Micro Focus Runtime Engine bereitstellen und ausführen. Bei der CardDemo Beispielanwendung handelt es sich um eine vereinfachte Kreditkartenanwendung, die entwickelt wurde, um Technologien für Anwendungsfälle bei der Mainframe-Modernisierung zu testen AWS und zu präsentieren und zu präsentieren und mit Partnern zusammenzuarbeiten.

In diesem Tutorial erstellen Sie Ressourcen in anderen AWS-Services Bereichen. Dazu gehören Amazon Simple Storage Service, Amazon Relational Database Service AWS Key Management Service und AWS Secrets Manager.

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Einen Amazon S3 S3-Bucket erstellen und laden](#)
- [Schritt 2: Eine Datenbank erstellen und konfigurieren](#)
- [Schritt 3: Erstellen und konfigurieren Sie ein AWS KMS key](#)
- [Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel](#)
- [Schritt 5: Erstellen Sie eine Laufzeitumgebung](#)
- [Schritt 6: Erstellen Sie eine Anwendung](#)
- [Schritt 7: Stellen Sie eine Anwendung bereit](#)
- [Schritt 8: Datensätze importieren](#)

- [Schritt 9: Starten Sie eine Anwendung](#)
- [Schritt 10: Connect zur CardDemo CICS-Anwendung her](#)
- [Bereinigen von -Ressourcen](#)
- [Nächste Schritte](#)

## Voraussetzungen

- Stellen Sie sicher, dass Sie Zugriff auf einen 3270-Emulator haben, um die CICS-Verbindung verwenden zu können. Kostenlose 3270-Emulatoren und Testversionen sind auf Websites von Drittanbietern erhältlich. Alternativ können Sie eine AWS Mainframe Modernization AppStream 2.0 Micro Focus-Instanz starten und den Rumba 3270-Emulator verwenden (nicht kostenlos erhältlich).

Informationen AppStream zu 2.0 finden Sie unter [the section called “Tutorial: Richten Sie AppStream 2.0 für Enterprise Analyzer und Enterprise Developer ein”](#)

### Note

Wählen Sie beim Erstellen des Stacks die Option Enterprise Developer (ED) und nicht Enterprise Analyzer (EA).

- Laden Sie die [CardDemo Beispielanwendung](#) herunter und entpacken Sie die heruntergeladene Datei in ein beliebiges lokales Verzeichnis. Dieses Verzeichnis wird ein Unterverzeichnis mit dem Titel enthalten. CardDemo
- Identifizieren Sie eine VPC in Ihrem Konto, in der Sie die in diesem Tutorial erstellten Ressourcen definieren können. Die VPC benötigt Subnetze in mindestens zwei Availability Zones. Weitere Informationen zu Amazon VPC finden Sie unter [So funktioniert Amazon VPC](#).

## Schritt 1: Einen Amazon S3 S3-Bucket erstellen und laden

In diesem Schritt erstellen Sie einen Amazon S3 S3-Bucket und laden CardDemo Dateien in diesen Bucket hoch. Später in diesem Tutorial verwenden Sie diese Dateien, um die CardDemo Beispielanwendung in einer Micro Focus Managed Runtime-Umgebung für die AWS Mainframe-Modernisierung bereitzustellen und auszuführen.

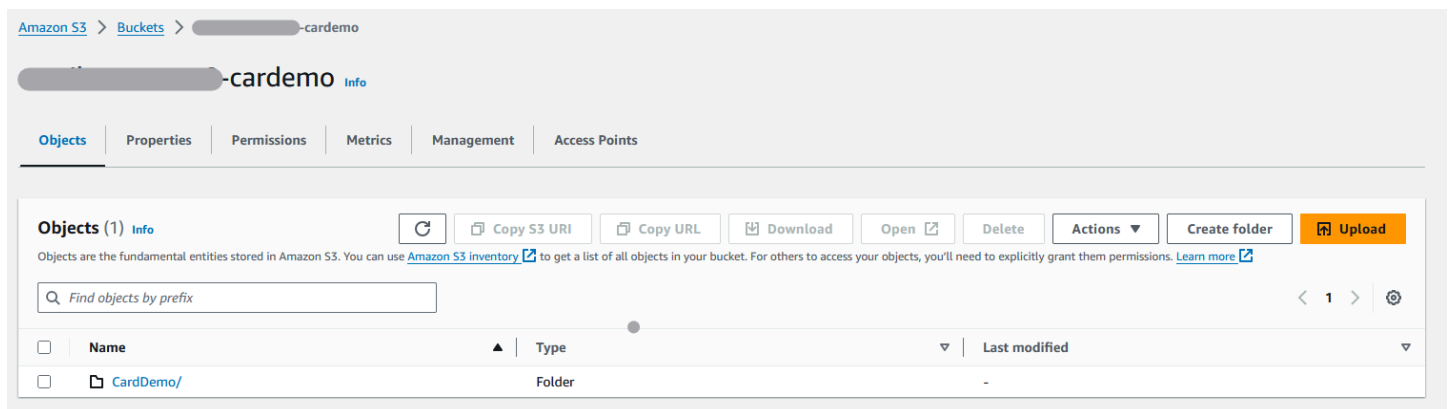
**Note**

Sie müssen keinen neuen S3-Bucket erstellen, aber der von Ihnen gewählte Bucket muss sich in derselben Region befinden wie die anderen in diesem Tutorial verwendeten Ressourcen.

## So erstellen Sie einen Amazon-S3-Bucket

1. Öffnen Sie die [Amazon S3 S3-Konsole](#) und wählen Sie Create Bucket aus.
2. Wählen Sie unter Allgemeine Konfiguration die AWS-Region aus, in der Sie die AWS Mainframe Modernization Micro Focus Managed Runtime erstellen möchten.
3. Geben Sie einen Bucket-Namen ein, zum Beispiel. `yourname-aws-region-carddemo`. Behalten Sie die Standardeinstellungen bei und wählen Sie Create Bucket. Alternativ können Sie auch Einstellungen aus einem vorhandenen Amazon S3 S3-Bucket kopieren und dann Bucket erstellen wählen.
4. Wählen Sie den Bucket aus, den Sie gerade erstellt haben, und wählen Sie dann Upload.
5. Wählen Sie im Bereich Upload die Option Ordner hinzufügen aus und navigieren Sie dann von Ihrem lokalen Computer aus zu dem CardDemo Verzeichnis.
6. Wählen Sie Upload, um den Upload-Vorgang zu starten. Die Upload-Zeiten variieren je nach Verbindungsgeschwindigkeit.
7. Wenn der Upload abgeschlossen ist, vergewissern Sie sich, dass alle Dateien erfolgreich hochgeladen wurden, und wählen Sie dann Schließen.

Ihr Amazon S3 S3-Bucket enthält jetzt den CardDemo Ordner.



The screenshot shows the Amazon S3 console interface for a bucket named 'carddemo'. The breadcrumb navigation at the top indicates the path: Amazon S3 > Buckets > [redacted]-carddemo. Below the bucket name, there are tabs for Objects, Properties, Permissions, Metrics, Management, and Access Points. The 'Objects' tab is active, showing a list of objects. At the top of the object list, there are several action buttons: Refresh, Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload. Below these buttons is a search bar with the placeholder text 'Find objects by prefix'. The object list has columns for Name, Type, and Last modified. One object is listed: 'CardDemo/' with a folder icon and the type 'Folder'.



Informationen zu S3-Buckets finden Sie unter [Amazon S3 S3-Buckets erstellen, konfigurieren und damit arbeiten](#).

## Schritt 2: Eine Datenbank erstellen und konfigurieren

In diesem Schritt erstellen Sie eine PostgreSQL-Datenbank in Amazon Relational Database Service (Amazon RDS). Für das Tutorial enthält diese Datenbank die Datensätze, die die CardDemo Beispielanwendung für Kundenaufgaben im Zusammenhang mit Kreditkartentransaktionen verwendet.

So erstellen Sie eine Datenbank in Amazon RDS

1. Öffnen Sie die [Amazon RDS-Konsole](#).
2. Wählen Sie die AWS-Region aus, in der Sie die Datenbank-Instance erstellen möchten.
3. Wählen Sie im Navigationsbereich Databases (Datenbanken) aus.
4. Wählen Sie Datenbank erstellen und dann Standard erstellen aus.
5. Wählen Sie als Engine-Typ PostgreSQL aus.
6. Wählen Sie eine Engine-Version von 15 oder höher.

### Note

Speichern Sie die Engine-Version, da Sie sie später in diesem Tutorial benötigen.

7. Wählen Sie unter Templates (Vorlagen) die Option Free tier (Kostenloses Kontingent) aus.
8. Ändern Sie die DB-Instance-ID in einen aussagekräftigen Wert, `MicroFocus-Tutorial` z. B.
9. Unterlassen Sie die Verwaltung von Master-Anmeldeinformationen in AWS Secrets Manager. Geben Sie stattdessen ein Master-Passwort ein und bestätigen Sie es.

### Note

Speichern Sie den Benutzernamen und das Passwort, die Sie für die Datenbank verwenden. Sie werden sie in den nächsten Schritten dieses Tutorials sicher speichern.

10. Wählen Sie unter Konnektivität die VPC aus, in der Sie die von AWS Mainframe Modernization verwaltete Laufzeitumgebung erstellen möchten.
11. Wählen Sie Datenbank erstellen aus.

## So erstellen Sie eine benutzerdefinierte Parametergruppe in Amazon RDS

1. Wählen Sie im Navigationsbereich der Amazon RDS-Konsole Parametergruppen und dann Parametergruppe erstellen aus.
2. Wählen Sie im Fenster Parametergruppe erstellen für Parametergruppenfamilie die Postgres-Option aus, die Ihrer Datenbankversion entspricht.

### Note

Für einige Postgres-Versionen ist ein Typ erforderlich. Wählen Sie bei Bedarf DB-Parametergruppe aus. Geben Sie einen Gruppennamen und eine Beschreibung für die Parametergruppe ein.

3. Wählen Sie Erstellen.

## Um die benutzerdefinierte Parametergruppe zu konfigurieren

1. Wählen Sie die neu erstellte Parametergruppe aus.
2. Wählen Sie Actions und anschließend Bearbeiten.
3. Filtern Sie nach `max_prepared_transactions` und ändern Sie den Parameterwert auf 100.
4. Wählen Sie Save Changes.

## Um die benutzerdefinierte Parametergruppe mit der Datenbank zu verknüpfen

1. Wählen Sie im Navigationsbereich der Amazon RDS-Konsole Datenbanken und dann die Datenbank-Instance aus, die Sie ändern möchten.
2. Wählen Sie Modify aus. Die Seite Modify DB instance (DB-Instance ändern) wird angezeigt.

### Note

Die Option Ändern ist erst verfügbar, wenn die Erstellung und Sicherung der Datenbank abgeschlossen ist, was mehrere Minuten dauern kann.

3. Navigieren Sie auf der Seite DB-Instance modifizieren zu Zusätzliche Konfiguration und ändern Sie die DB-Parametergruppe in Ihre Parametergruppe. Wenn Ihre Parametergruppe nicht in der Liste verfügbar ist, überprüfen Sie, ob sie mit der richtigen Datenbankversion erstellt wurde.
4. Wählen Sie Weiter und überprüfen Sie die Zusammenfassung der Änderungen.

5. Wählen Sie Sofort anwenden, um die Änderungen sofort zu übernehmen.
6. Klicken Sie auf DB-Instance ändern, um Ihre Änderungen zu speichern.

Weitere Informationen zu Parametergruppen finden Sie unter [Mit Parametergruppen arbeiten](#).

#### Note

Sie können auch eine Amazon Aurora PostgreSQL-Datenbank mit AWS Mainframe-Modernisierung verwenden, es gibt jedoch keine kostenlose Kontingentoption. Weitere Informationen finden Sie unter [Arbeiten mit Amazon Aurora PostgreSQL](#).

## Schritt 3: Erstellen und konfigurieren Sie ein AWS KMS key

Um Anmeldeinformationen für die Amazon RDS-Instance sicher zu speichern, erstellen Sie zunächst eine AWS KMS key.

Um eine zu erstellen AWS KMS key

1. Öffnen Sie die [Key Management Service-Konsole](#).
2. Klicken Sie auf Create key (Schlüssel erstellen).
3. Behalten Sie die Standardwerte Symmetrisch für den Schlüsseltyp und Verschlüsseln und Entschlüsseln für die Schlüsselverwendung bei.
4. Wählen Sie Weiter aus.
5. Geben Sie dem Schlüssel einen Alias wie MicroFocus-Tutorial-RDS-Key und eine optionale Beschreibung.
6. Wählen Sie Weiter aus.
7. Weisen Sie einen Schlüsseladministrator zu, indem Sie das Kästchen neben Ihrem Benutzer oder Ihrer Rolle ankreuzen.
8. Wählen Sie Weiter und dann erneut Weiter.
9. Bearbeiten Sie auf dem Überprüfungsbildschirm die Schlüsselrichtlinie und geben Sie dann Folgendes ein:

```
{  
  "Sid" : "Allow access for Mainframe Modernization Service",  
  "Effect" : "Allow",
```

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

Diese Richtlinie gewährt AWS Mainframe Modernization mithilfe dieser speziellen Schlüsselrichtlinie Entschlüsselungsberechtigungen.

10. Wählen Sie Fertig stellen, um den Schlüssel zu erstellen.

Weitere Informationen finden Sie unter [Schlüssel erstellen](#) im AWS Key Management Service Entwicklerhandbuch.

## Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel

Speichern Sie nun die Datenbankanmeldedaten sicher mit AWS Secrets Manager und AWS KMS key.

Um ein AWS Secrets Manager Datenbankgeheimnis zu erstellen und zu konfigurieren

1. Öffnen Sie die [Secrets Manager-Konsole](#).
2. Wählen Sie im Navigationsbereich Secrets (Geheimnisse).
3. Wählen Sie unter Secrets die Option Neues Geheimnis speichern aus.
4. Stellen Sie den Geheimtyp auf Anmeldeinformationen für die Amazon RDS-Datenbank ein.
5. Geben Sie die Anmeldeinformationen ein, die Sie bei der Erstellung der Datenbank angegeben haben.
6. Wählen Sie unter Verschlüsselungsschlüssel den Schlüssel aus, den Sie in Schritt 3 erstellt haben.
7. Wählen Sie im Abschnitt Datenbank die Datenbank aus, die Sie für dieses Tutorial erstellt haben, und klicken Sie dann auf Weiter.
8. Geben Sie unter Geheimer Name einen Namen wie MicroFocus-Tutorial-RDS-Secret und optional eine Beschreibung ein.
9. Wählen Sie im Abschnitt Ressourcenberechtigungen die Option Berechtigungen bearbeiten aus und ersetzen Sie den Inhalt durch die folgende Richtlinie:

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

10. Wählen Sie Speichern.
11. Wählen Sie „Weiter“ für die nachfolgenden Bildschirme und anschließend „Speichern“. Aktualisieren Sie die Geheimnisliste, um das neue Geheimnis zu sehen.
12. Wählen Sie das neu erstellte Geheimnis aus und notieren Sie sich das, Secret ARN weil Sie es später im Tutorial benötigen.
13. Wählen Sie auf der Registerkarte „Übersicht“ des Geheimnisses die Option Geheimes Wert abrufen aus.
14. Wählen Sie „Bearbeiten“ und anschließend „Zeile hinzufügen“.
15. Fügen Sie einen Schlüssel für sslMode mit einem Wert von `hinzuverify-full`:

#### Edit secret value

Key/value

Plaintext

sslMode

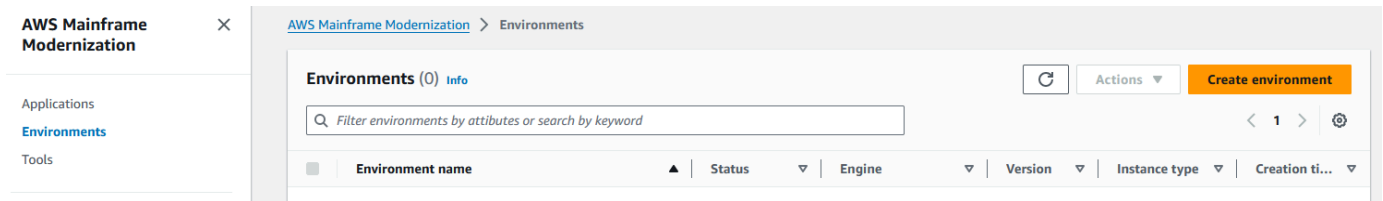
verify-full

16. Wählen Sie Speichern.

## Schritt 5: Erstellen Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung zu erstellen

1. Öffnen Sie die [AWS Mainframe Modernization Console](#).
2. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus. Wählen Sie dann Umgebung erstellen.



### 3. Unter Basisinformationen angeben

- a. Geben Sie MicroFocus-Environment den Namen der Umgebung ein.
- b. Vergewissern Sie sich, dass unter Engine-Optionen Micro Focus ausgewählt ist.
- c. Wählen Sie die neueste Micro Focus-Version.
- d. Wählen Sie Weiter aus.

## Name and description [Info](#)

### Environment name

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

### Environment description - *optional*

The description can be up to 500 characters.

## Engine options [Info](#)

### Select Engine

 Blu Age

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

 Micro Focus

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



### Micro Focus Version

4. Konfigurieren Sie die Umgebung
  - a. Wählen Sie unter Verfügbarkeit die Option Hochverfügbarkeitscluster aus.

- b. Wählen Sie unter Ressourcen entweder `M2.c5.large` oder `M2.m5.large` für den Instanztyp und die Anzahl der gewünschten Instanzen aus. Geben Sie bis zu zwei Instanzen an.
- c. Wählen Sie unter Sicherheit und Netzwerk die Option Öffentlichen Zugriff auf in dieser Umgebung bereitgestellte Anwendungen zulassen aus und wählen Sie mindestens zwei öffentliche Subnetze aus.
- d. Wählen Sie Weiter aus.

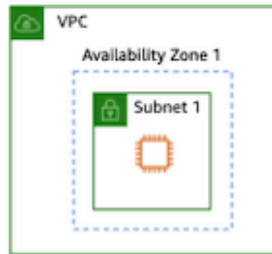


# Specify configurations [Info](#)

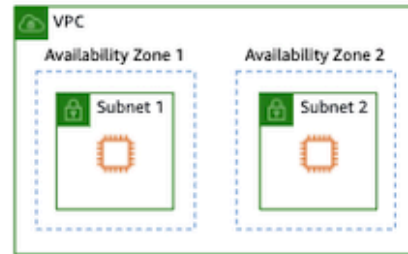
## Availability [Info](#)

Choose the availability pattern for your environment.

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



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



## Resources

### Instance type

Choose the instance type for your high availability cluster.

M2.m5.large

### Desired capacity

Specify the desired number of instances.

2

## Security and network

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

### Virtual Private Cloud (VPC)

Choose the VPC where you want to create the environment.

Default vpc-15

### Subnets

Choose one or more subnets for a high availability setup.

Choose subnets

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

subnet-6685 | us-west-2b X

### Security groups

Choose one or more security groups for the chosen VPC.

5. Wählen Sie auf der Seite Speicher anhängen die Option Weiter aus.
6. Wählen Sie auf der Seite Wartung planen die Option Keine Präferenz und dann Weiter aus.

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

**Maintenance window** [Info](#)

Select the period you want pending modifications or maintenance to be applied.

**When to apply modifications**

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

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

Cancel Previous **Next**

7. Überprüfen Sie auf der Seite Überprüfen und erstellen alle Konfigurationen, die Sie für die Laufzeitumgebung angegeben haben, und wählen Sie dann Umgebung erstellen aus.

### Step 3: Attach storage Edit

#### EFS storage

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

#### FSx storage

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

### Step 4: Schedule maintenance Edit

#### Maintenance window

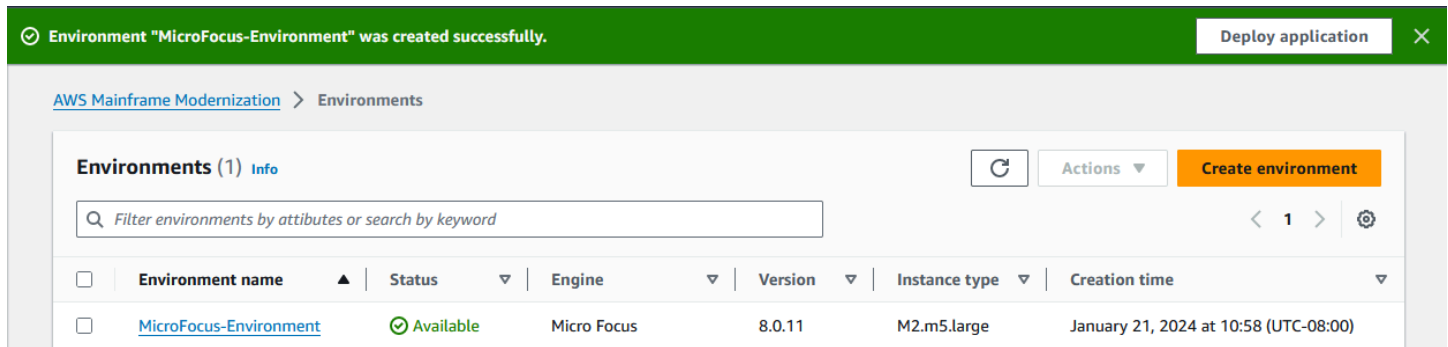
Preferred maintenance window  
No preference

Cancel Previous Create environment

Wenn Sie Ihre Umgebung erstellt haben, erscheint ein Banner mit der Aufschrift `Environment name was created successfully`, und das Feld Status ändert sich zu Verfügbar. Die Erstellung der Umgebung dauert einige Minuten, aber Sie können mit den nächsten Schritten fortfahren, während sie ausgeführt wird.

Schritt 5: Erstellen Sie eine Laufzeitumgebung

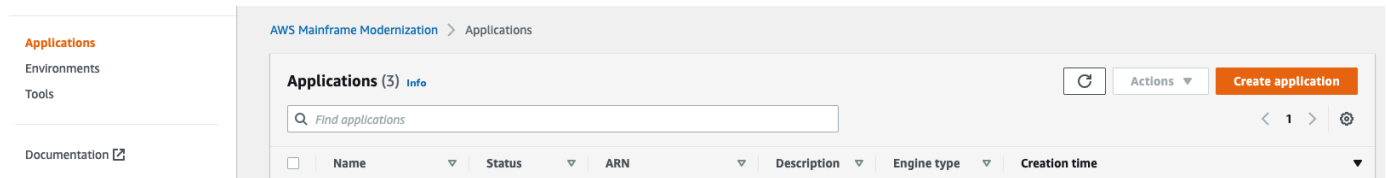
37



## Schritt 6: Erstellen Sie eine Anwendung

So erstellen Sie eine Anwendung

1. Wählen Sie im Navigationsbereich Applications (Anwendungen). Wählen Sie dann Anwendung erstellen.



2. Geben Sie auf der Seite Anwendung erstellen unter Basisinformationen angeben den MicroFocus-CardDemo Namen der Anwendung ein und stellen Sie sicher, dass unter Engine-Typ Micro Focus ausgewählt ist. Wählen Sie anschließend Weiter.

AWS Mainframe Modernization > Applications > Create application

Step 1  
**Specify basic information**

Step 2  
Specify resources and configurations

Step 3  
Review and create

## Specify basic information [Info](#)

### Name and description

Application name

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


Application description - *optional*

*Describe the application*


The maximum length is 500 characters.

### Engine type

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



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



3. Wählen Sie unter Ressourcen und Konfigurationen angeben die Option aus, um die Anwendungsdefinition mit ihren Ressourcen und Konfigurationen mithilfe des Inline-Editors anzugeben.

AWS Mainframe Modernization > Applications > Create application

Step 1  
[Specify basic information](#)

Step 2  
**Specify resources and configurations**

Step 3  
Review and create

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

### Resources and configurations

Choose an approach to define the application

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

1 {}

```
JSON Ln 1, Col 1 ❌ Errors: 0 ⚠️ Warnings: 0
```

The maximum size of the JSON file is 500 kB.

Cancel Previous **Next**

Geben Sie die folgende Anwendungsdefinition in den Editor ein:

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

```

],
"definition": {
  "listeners": [
    {
      "port": 6000,
      "type": "tn3270"
    }
  ],
  "dataset-location": {
    "db-locations": [
      {
        "name": "Database1",
        "secret-manager-arn":
"arn:aws:secretsmanager:Region:123456789012:secret:MicroFocus-Tutorial-RDS-Secret-
xxxxxxx"
      }
    ]
  },
  "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

Diese Datei kann sich ändern.

4. Bearbeiten Sie den Anwendungs-JSON im Eigenschaftenobjekt von source-locations wie folgt:
  - a. Ersetzen Sie den Wert für `s3_bucket` durch den Namen des Amazon S3 S3-Buckets, den Sie in Schritt 1 erstellt haben.
  - b. Ersetzen Sie den Wert für `s3-key-prefix` durch den Ordner (key prefix), in den Sie die CardDemo Beispieldateien hochgeladen haben. Wenn Sie das CardDemo Verzeichnis direkt in einen Amazon S3 S3-Bucket hochgeladen haben, muss `s3-key-prefix` das nicht geändert werden.
  - c. Ersetzen Sie beide `secret-manager-arn` Werte durch den ARN für das Datenbankgeheimnis, das Sie in Schritt 4 erstellt haben.



## Resources and configurations

Choose an approach to define the application

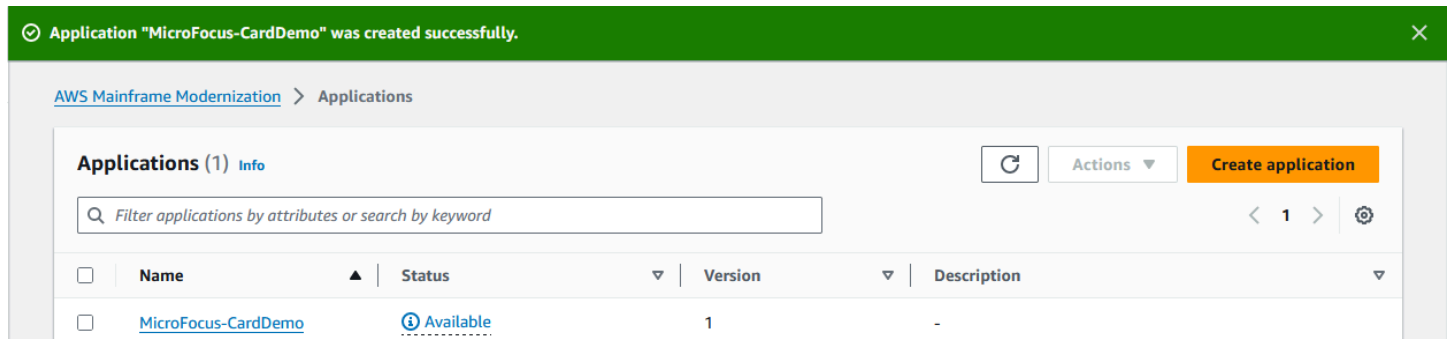
- Specify the application definition with its resources and configurations using the inline editor
- Use an application definition JSON file in an Amazon S3 bucket

```
1 {
2   "template-version": "2.0",
3   "source-locations": [
4     {
5       "source-id": "s3-source",
6       "source-type": "s3",
7       "properties": {
8         "s3-bucket": "XXXXXXXXXXXX-cardemo",
9         "s3-key-prefix": "CardDemo"
10      }
11    }
12  ],
13  "definition": {
14    "listeners": [{"id": "listener1"}],
15    "dataset-location": {
16      "db-locations": [
17        {
18          "name": "Database1",
19          "secret-manager-arn": "arn:aws:secretsmanager:XXXXXXXXXXXX:secret/XXXXXXXXXXXX"
20        }
21      ]
22    }
23  },
24  "batch-settings": {
25  }
26 }
27 }
28 }
29 }
```

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

Weitere Informationen zur Anwendungsdefinition finden Sie unter [Micro Focus-Anwendungsdefinition](#).

5. Wählen Sie Next (Weiter), um fortzufahren.
6. Überprüfen Sie auf der Seite Überprüfen und erstellen die von Ihnen bereitgestellten Informationen, und wählen Sie dann Anwendung erstellen aus.

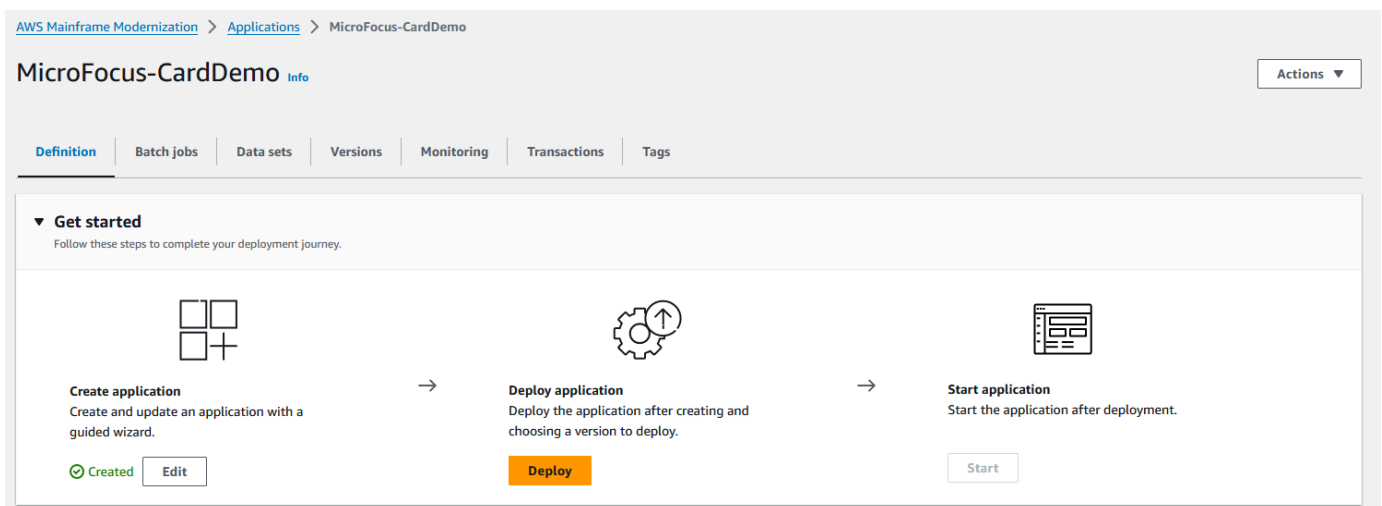


Wenn Sie Ihre Anwendung erstellt haben, erscheint ein Banner mit der Aufschrift *Application name* was created successfully. Und das Statusfeld ändert sich zu Verfügbar.

## Schritt 7: Stellen Sie eine Anwendung bereit

Um eine Anwendung bereitzustellen

1. Wählen Sie im Navigationsbereich Anwendungen und dann *MicroFocus-CardDemo*.
2. Wählen Sie unter Anwendung bereitstellen die Option Bereitstellen aus.



3. Wählen Sie die neueste Version der Anwendung und die Umgebung, die Sie zuvor erstellt haben, und wählen Sie dann Deploy aus.

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

## Deploy application Info

You have selected the following application:

Name	Description	Engine
MicroFocus-CardDemo	-	Micro Focus

**Available versions (1/1)** ↻

Choose a version from the list.

< 1 > ⚙️

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

**Environments (1/1) Info**

< 1 > ⚙️

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

Cancel Deploy

Wenn die CardDemo Anwendung erfolgreich bereitgestellt wurde, ändert sich der Status in Bereit.

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

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

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

< 1 > ⚙️

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

## Schritt 8: Datensätze importieren

Um Datensätze zu importieren

1. Wählen Sie im Navigationsbereich Anwendungen und dann die Anwendung aus.
2. Wählen Sie die Registerkarte Datensätze aus. Wählen Sie anschließend Import (Importieren) aus.
3. Wählen Sie JSON-Konfiguration importieren und bearbeiten und wählen Sie dann die Option Eigenes JSON kopieren und einfügen.

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

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

Choose import method.

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

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

#### JSON configuration

Import from Amazon S3 bucket.

Copy and paste your own JSON.

1		
---	--	--

4. Kopieren Sie den folgenden JSON-Code und fügen Sie ihn ein, wählen Sie aber noch nicht „Senden“. Diese JSON-Datei enthält alle für die Demo-Anwendung erforderlichen Datensätze, benötigt jedoch Ihre Amazon S3 S3-Bucket-Details.

```
{
  "dataSets": [
    {
      "dataSet": {
```

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

```

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

```

```

        "recordLength": {
            "min": 50,
            "max": 50
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS.DAT"
    }
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",
                "primaryKey": {
                    "length": 9,
                    "offset": 0
                }
            }
        },
        "recordLength": {
            "min": 500,
            "max": 500
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CUSTDATA.VSAM.KSDS.DAT"
    }
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.CARDXREF.VSAM.AIX.PATH",
        "relativePath": "DATA",
        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",

```

```
        "primaryKey": {
            "length": 11,
            "offset": 25
        }
    },
    "recordLength": {
        "min": 50,
        "max": 50
    }
},
"externalLocation": {
    "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.CARDXREF.VSAM.KSDS.DAT"
}
},
{
    "dataSet": {
        "storageType": "Database",
        "datasetName": "AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS",
        "relativePath": "DATA",
        "datasetOrg": {
            "vsam": {
                "format": "KS",
                "encoding": "A",
                "primaryKey": {
                    "length": 16,
                    "offset": 0
                }
            }
        },
        "recordLength": {
            "min": 350,
            "max": 350
        }
    },
    "externalLocation": {
        "s3Location": "s3://<s3-bucket-name>/CardDemo/catalog/data/
AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT"
    }
},
{
    "dataSet": {
        "storageType": "Database",
```



```

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

```

- Ersetzen Sie jedes Vorkommen von `<s3-bucket-name>` (es gibt acht) durch den Namen des Amazon S3 S3-Buckets, der den CardDemo Ordner enthält, zum Beispielyour-name-aws-region-carddemo.

#### Note

Um die Amazon S3-URI für den Ordner in Amazon S3 zu kopieren, wählen Sie den Ordner aus und wählen Sie dann Amazon S3 S3-URI kopieren.

- Wählen Sie Absenden aus.

Wenn der Import abgeschlossen ist, erscheint ein Banner mit der folgenden Meldung: Import task with resource identifier *name* was completed successfully. Eine Liste der importierten Datensätze wird angezeigt.

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

AWS Mainframe Modernization > Applications > MicroFocus-CardDemo

## MicroFocus-CardDemo Info

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

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

Filter data sets by name

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

Sie können auch den Status aller Datensatz-Importe anzeigen, indem Sie auf der Registerkarte Datensätze die Option Importverlauf auswählen.

## Schritt 9: Starten Sie eine Anwendung

Um eine Anwendung zu starten


1. Wählen Sie im Navigationsbereich Anwendungen und dann die Anwendung aus.
2. Wählen Sie Anwendung starten aus.

AWS Mainframe Modernization > Applications > MicroFocus-CardDemo

## MicroFocus-CardDemo Info


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

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




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

✔ Created Edit



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

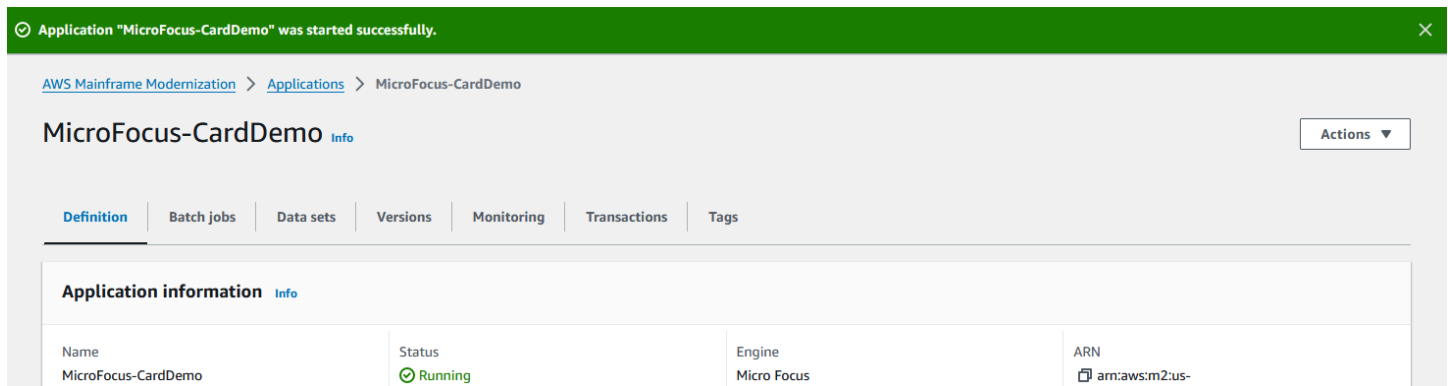
✔ Deployed Deploy



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

⊖ Stopped Start

Wenn die CardDemo Anwendung erfolgreich ausgeführt wird, erscheint ein Banner mit der folgenden Meldung: `Application name was started successfully`. Das Statusfeld ändert sich zu `Wird ausgeführt`.



## Schritt 10: Connect zur CardDemo CICS-Anwendung her

Bevor Sie eine Verbindung herstellen, stellen Sie sicher, dass die VPC und die Sicherheitsgruppe, die Sie für die Anwendung angegeben haben, mit denen identisch sind, die Sie für Ihre Netzwerkschnittstelle beantragt haben, von der aus Sie eine Verbindung herstellen werden.

Um die TN3270-Verbindung zu konfigurieren, benötigen Sie außerdem den DNS-Hostnamen und den Port der Anwendung.

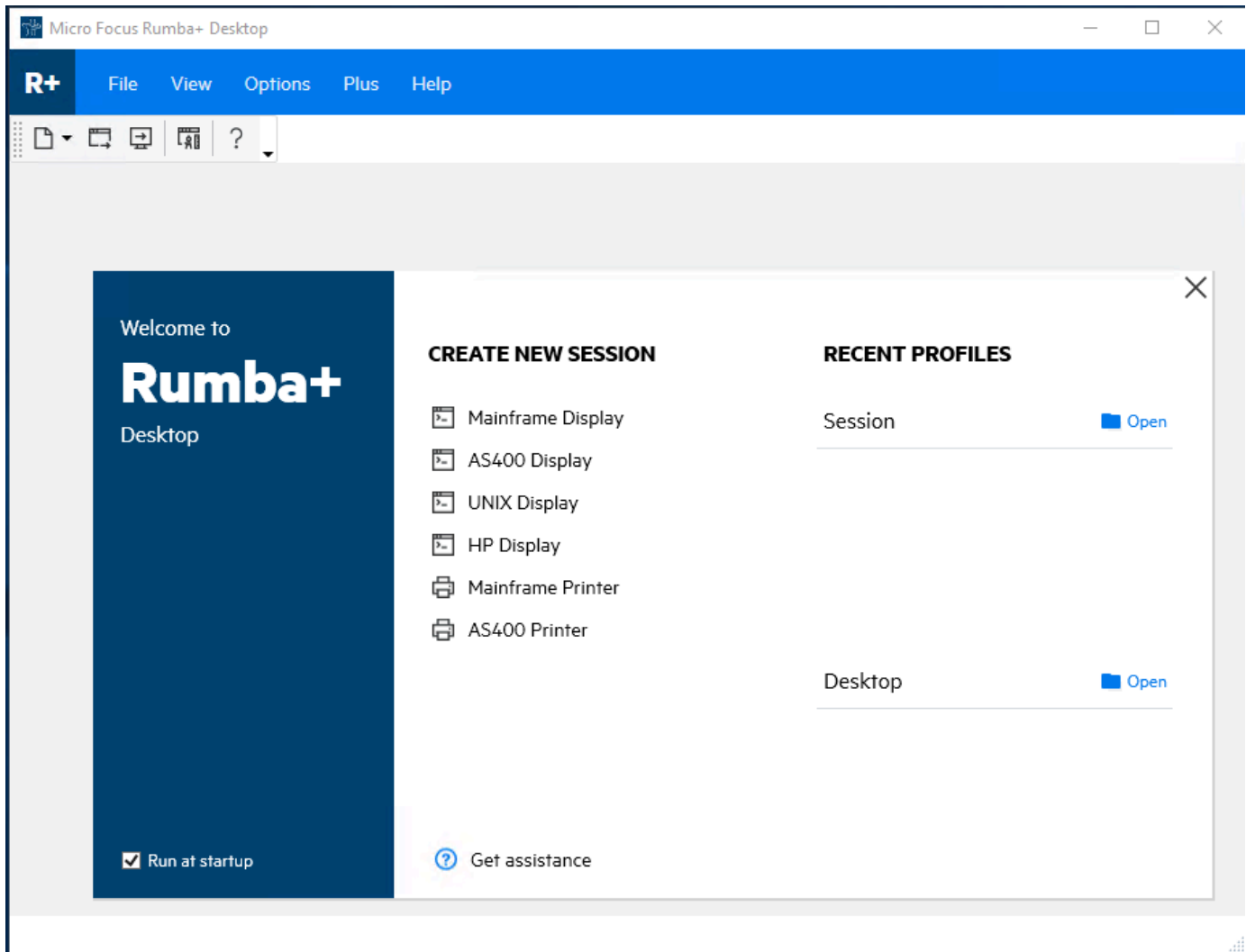
Um eine Anwendung mithilfe des Terminalemulators zu konfigurieren und mit dem Mainframe zu verbinden

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole und wählen Sie Applications und anschließend. `MicroFocus-CardDemo`
2. Wählen Sie das Kopiersymbol, um den DNS-Hostnamen zu kopieren. Notieren Sie sich auch die Port-Nummer.
3. Starten Sie einen Terminalemulator. In diesem Tutorial wird Micro Focus Rumba+ verwendet.

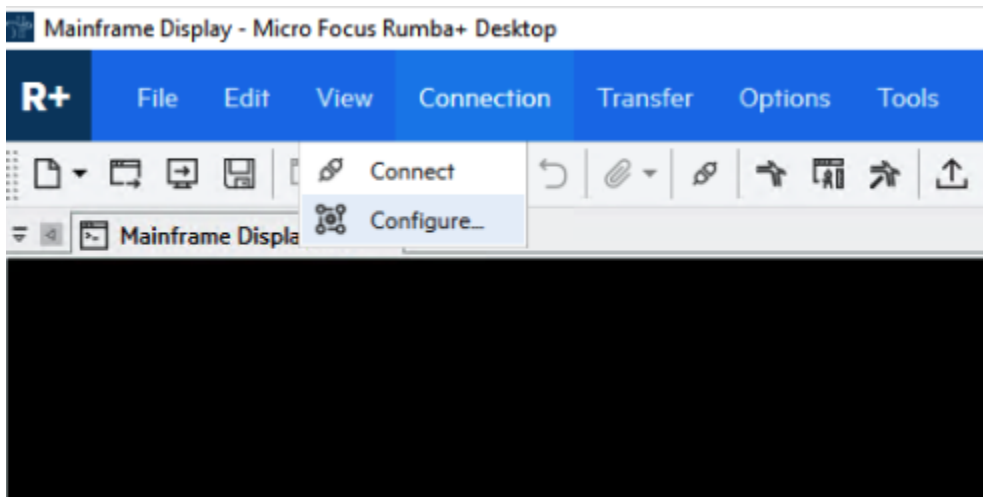
### Note

Die Konfigurationsschritte variieren je nach Emulator.

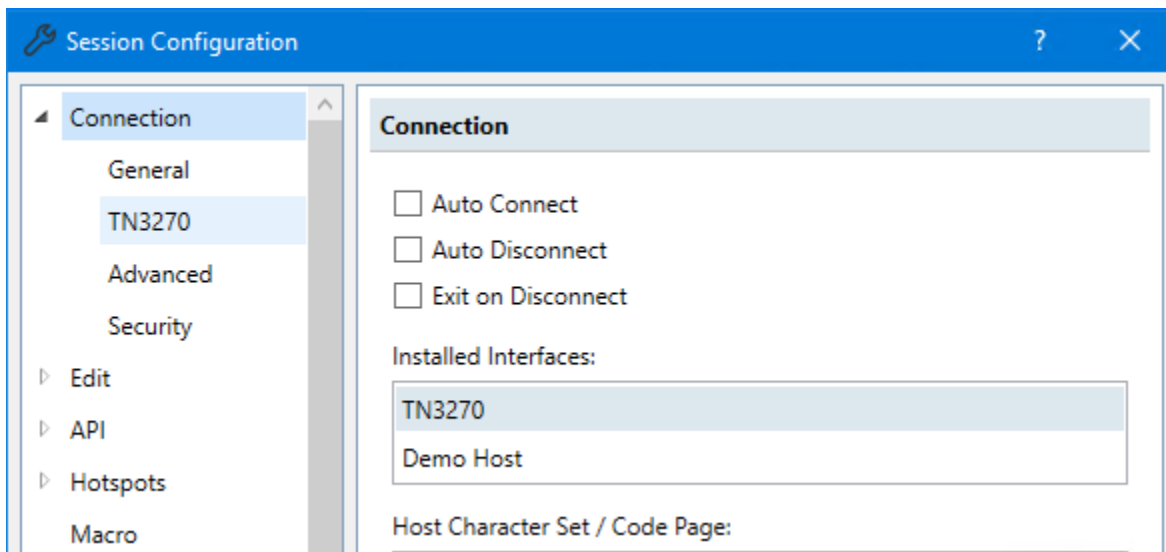
4. Wählen Sie Mainframe Display.



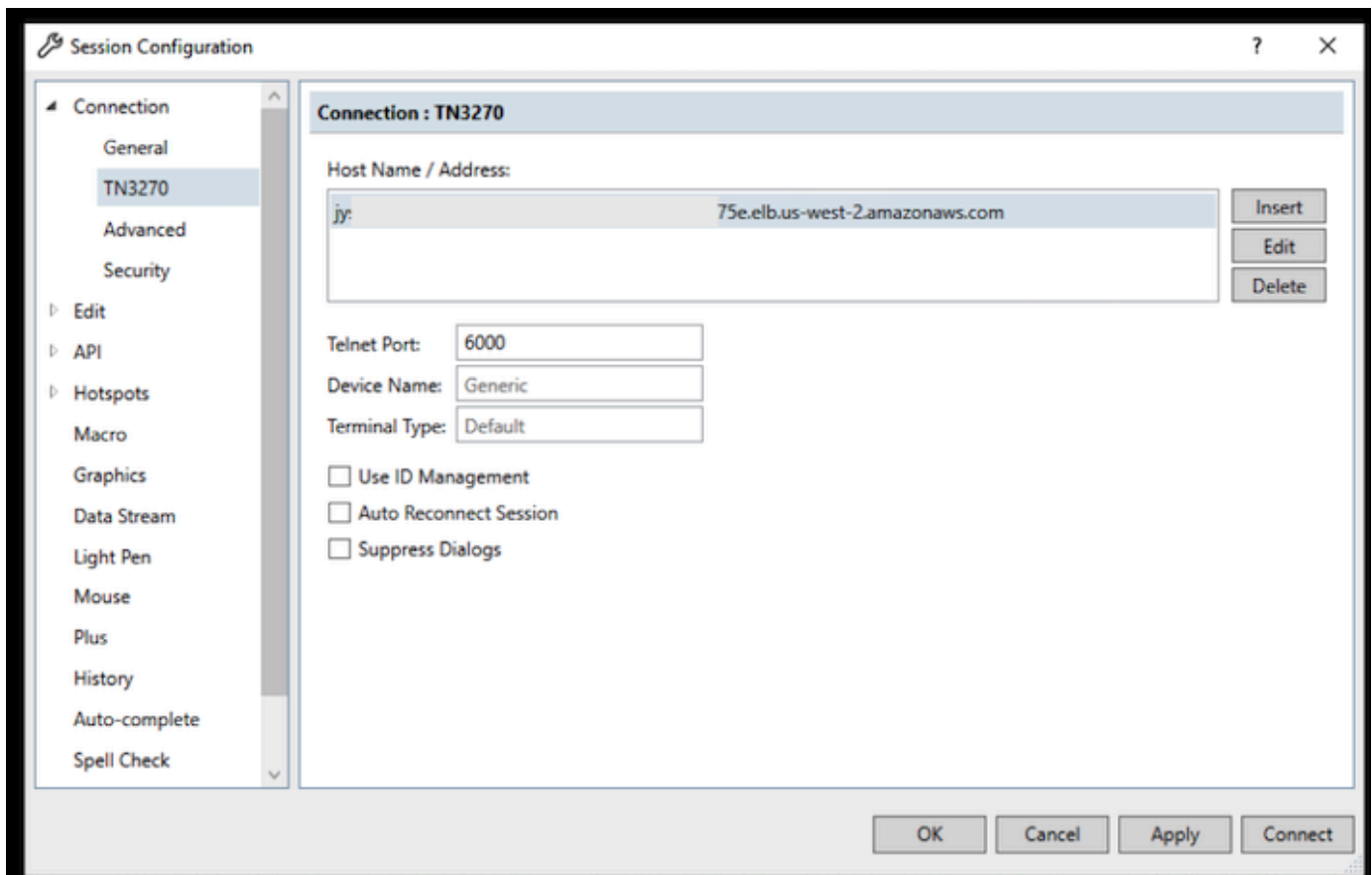
5. Wählen Sie „Verbindung“ und anschließend „Konfigurieren“.




6. Wählen Sie unter Installierte Schnittstellen die Option ausTN3270, und wählen Sie dann TN3270 erneut im Verbindungsmenü.




7. Wählen Sie Einfügen und fügen Sie das DNS Hostname für die Anwendung ein. Geben Sie 6000 für den Telnet-Port an.



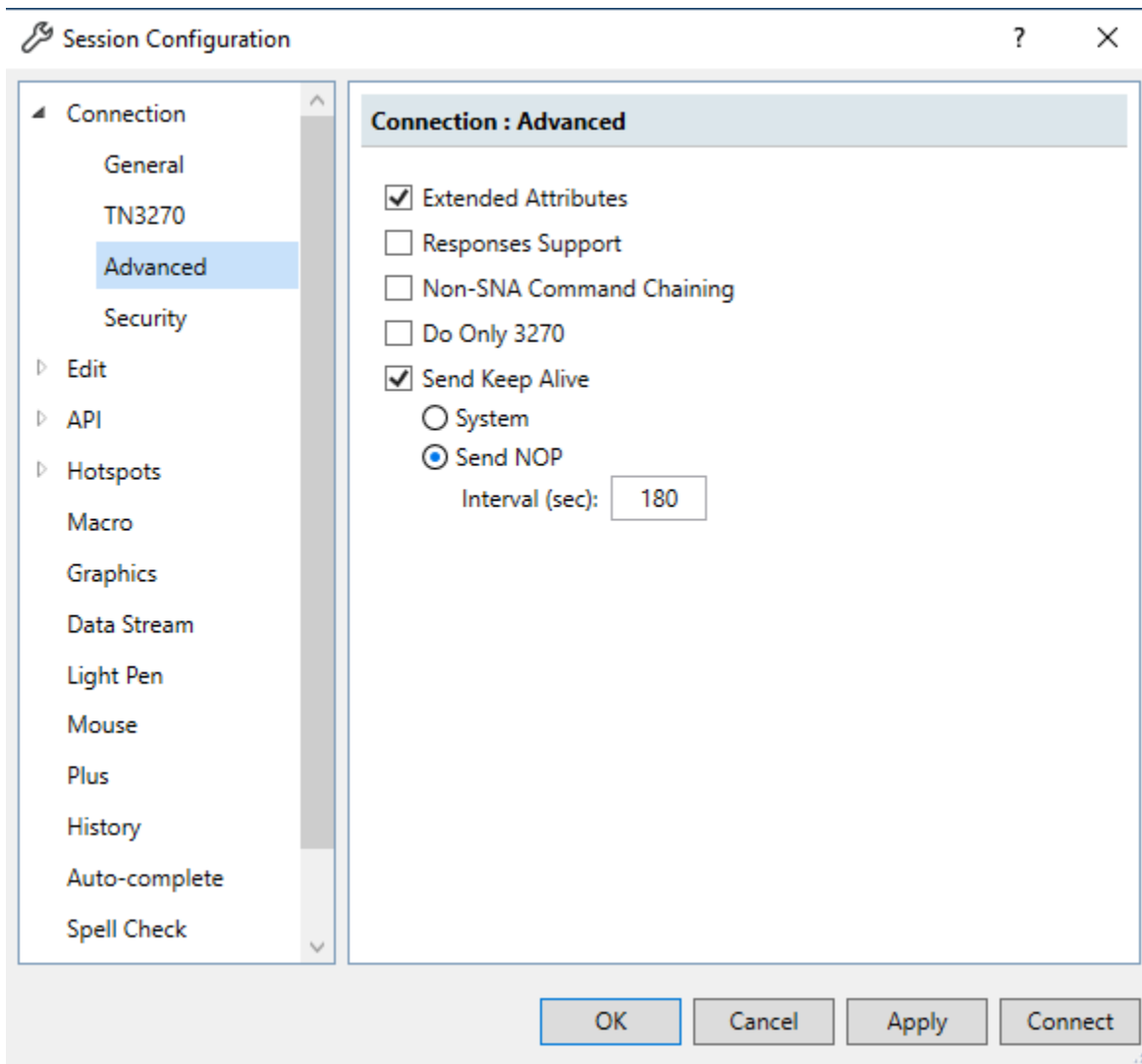
 Note

Wenn Sie AWS AppStream 2.0 in einem Browser verwenden und Probleme beim Einfügen von Werten haben, finden Sie weitere Informationen unter [Problembehandlung für AppStream 2.0-Benutzerprobleme](#).

8. Wählen Sie unter Verbindung die Option Erweitert und dann Send Keep Alive und Send NOP aus und geben Sie 180 als Intervall ein.

 Note

Wenn Sie die Keep-Alive-Einstellung auf Ihrem TN3270-Terminal auf mindestens 180 Sekunden konfigurieren, können Sie sicherstellen, dass der Network Load Balancer Ihre Verbindung nicht unterbricht.



9. Wählen Sie Connect aus.


**Note**

Wenn die Verbindung fehlschlägt:

- Wenn Sie AppStream 2.0 verwenden, stellen Sie sicher, dass die für die Anwendungsumgebung angegebene VPC und Sicherheitsgruppe mit der 2.0-Flotte identisch sind. AppStream
- Verwenden Sie den VPC Reachability Analyzer, um die Verbindung zu analysieren. [Sie können über die Konsole auf den Reachability Analyzer zugreifen.](#)

- Versuchen Sie als Diagnoseschritt, die Regeln für eingehende Sicherheitsgruppen für das Programm hinzuzufügen oder zu ändern, um Datenverkehr über Port 6000 von überall her zuzulassen (d. h. CIDR Block 0.0.0.0/0). Wenn Sie erfolgreich eine Verbindung herstellen, wissen Sie, dass die Sicherheitsgruppe Ihren Datenverkehr blockiert hat. Ändern Sie die Quelle der Sicherheitsgruppe auf etwas Spezifischeres. Weitere Informationen zu Sicherheitsgruppen finden Sie unter [Grundlagen zu Sicherheitsgruppen](#).

10. Geben Sie USER0001 den Benutzernamen und password das Passwort ein.

 Note

In Rumba ist ctrl-shift-z die Standardeinstellung für Clear und die Standardeinstellung für Reset Strg-R.



```

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

This is a Credit Card Demo Application for Mainframe Modernization

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

Type your User ID and Password, then press ENTER:

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

ENTER=Sign-on F3=Exit

Ready | Running | APL | NUMFLD | NETB000 | OVR | CAP | NUM | W | 20.62 | gu3tqnjyqzdmml-2mj...

```

11. Nachdem Sie sich erfolgreich angemeldet haben, können Sie durch die CardDemo Anwendung navigieren.
12. Geben Sie 01 für die Kontoansicht ein.

```
Tran: CM00                      AWS Mainframe Modernization      Date: 01/22/24
Prog: COMEN01C                   CardDemo                          Time: 00:22:39

                               Main Menu

                               01. Account View
                               02. Account Update
                               03. Credit Card List
                               04. Credit Card View
                               05. Credit Card Update
                               06. Transaction List
                               07. Transaction View
                               08. Transaction Add
                               09. Transaction Reports
                               10. Bill Payment

                               Please select an option : 01

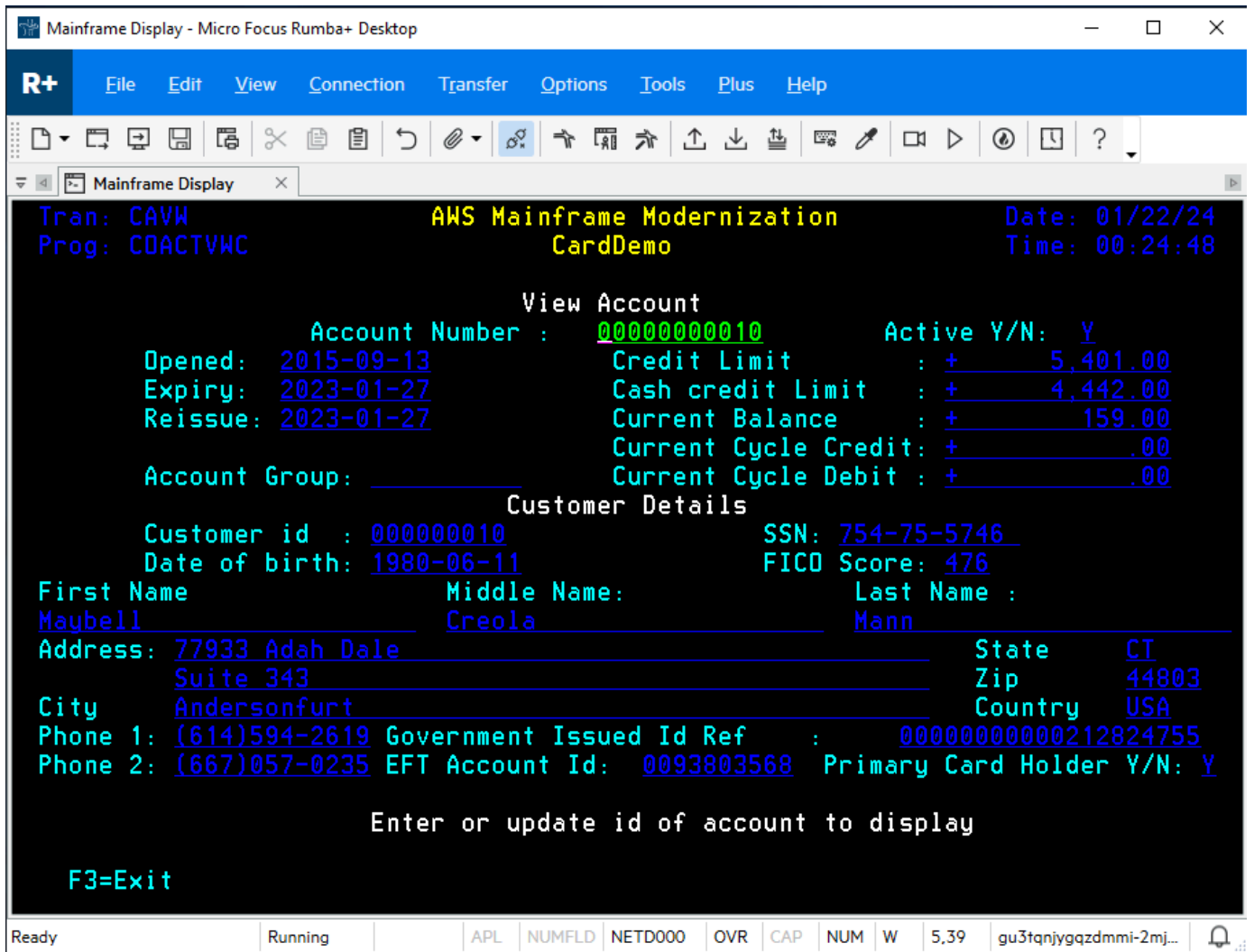
                               ENTER=Continue  F3=Exit
```

Ready | Running | APL | NUMFLD | NETD000 | OVR | CAP | NUM | W | 20.42 | gu3tanjyqazdmmi-2mj... | 🔔

13. Geben Sie 0000000010 die Kontonummer ein und drücken Sie die Eingabetaste auf Ihrer Tastatur.

**Note**

Andere gültige Konten sind 0000000011 und 0000000020.



14. Drücken Sie F3, um das Menü zu verlassen und die Transaktion F3 zu beenden.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, um zusätzliche Kosten zu vermeiden. Führen Sie dazu die folgenden Schritte aus:

- Falls erforderlich, beenden Sie die Anwendung.
- Löschen Sie die -Anwendung. Weitere Informationen finden Sie unter [Löschen Sie eine AWS Mainframe-Modernisierungsanwendung](#).
- Löschen Sie die Laufzeitumgebung. Weitere Informationen finden Sie unter [Löschen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#).

- Löschen Sie die Amazon S3 S3-Buckets, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.
- Löschen Sie das AWS Secrets Manager Geheimnis, das Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen eines Geheimnisses](#).
- Löschen Sie den KMS-Schlüssel, den Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen von AWS-KMS-Schlüsseln](#).
- Löschen Sie die Amazon RDS-Datenbank, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen der EC2-Instance und DB-Instance](#) im Amazon RDS-Benutzerhandbuch.
- Wenn Sie eine Sicherheitsgruppenregel für Port 6000 hinzugefügt haben, löschen Sie die Regel.

## Nächste Schritte

Informationen zum Einrichten einer Entwicklungsumgebung für Ihre modernisierten Anwendungen finden Sie unter [Tutorial: Setup AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer](#).

# Modernisierungsansatz

Die Migration ist komplex und hat viele Variablen. AWS Die Mainframe-Modernisierung bietet einen evolutionären Ansatz, der einige kurzfristige Vorteile bietet, indem er die Agilität verbessert und später zahlreiche Optimierungs- und Innovationsmöglichkeiten bietet. Darüber hinaus trägt die AWS Mainframe-Modernisierung dazu bei, die Umstellung zu vereinfachen und berücksichtigt dennoch die Besonderheiten des Unternehmens und der Geschäftstätigkeit Ihres Kunden. Die beiden Hauptansätze, die von der AWS Mainframe-Modernisierung unterstützt werden, sind automatisiertes Refactoring oder Replatforming. Welche Sie wählen sollten, hängt von der Situation Ihres Kunden ab.

Beim automatisierten Refactoring werden AWS Blu Age-Tools verwendet, um Code, Daten und Abhängigkeiten automatisch in moderne Sprachen, Datenspeicher und Frameworks zu konvertieren und gleichzeitig die funktionale Äquivalenz mit denselben Geschäftsfunktionen zu gewährleisten.

Beim Replatforming werden Tools von Micro Focus verwendet, um Mainframe-Workloads in agile Services umzuwandeln. AWS

Sie können sich den Modernisierungsprozess in Etappen vorstellen. Die erste Phase umfasst drei Phasen: Bewertung, Mobilisierung sowie Migration und Modernisierung. Die nächste Phase umfasst die Betriebs- und Optimierungsphase, in der Sie weitere Innovationsmöglichkeiten identifizieren können.

## Themen

- [Phase bewerten](#)
- [Phase der Mobilisierung](#)
- [Phase der Migration und Modernisierung](#)
- [Phase „Betrieb und Optimierung“](#)

## Phase bewerten

Auf der höchsten Ebene wird in der Bewertungsphase geprüft, ob Sie für die Migration bereit sind. Sie definieren ein Geschäftsszenario und schulen Ihr Team anschließend mit Workshops und einem Immersionstag (Demos und Übungen), der von angeboten wird. AWS In Workshops und Immersionstagen werden verschiedene Themen behandelt. Diese Aufgaben werden außerhalb der AWS Mainframe-Modernisierung durchgeführt.

## Phase der Mobilisierung

In der Mobilize-Phase beginnen Sie Ihr Projekt mit einem Kick-Off und durchlaufen dann einen Discovery-Prozess, bei dem Daten aus Ihren Mainframe-Anwendungen extrahiert und in ein Migrationstool eingespeist werden. Sie identifizieren die Anwendungen, die Sie migrieren möchten, und wählen einige Anwendungen für die Pilotphase aus. Sie verfeinern Ihren Geschäftsszenario, erstellen Ihren Migrationsplan und entscheiden, wie Sie mit Sicherheit und Compliance, Account-Governance und Ihrem Betriebsmodell umgehen möchten. Sie richten ein Cloud-Kompetenzzentrum mit den richtigen Mitarbeitern aus Ihrem Team ein. Sie führen die Pilotprojekte durch und dokumentieren, was Sie gelernt haben. Sie verfeinern Ihren Migrationsplan und Ihr Geschäftsszenario. Viele dieser Aufgaben werden außerhalb der AWS Mainframe-Modernisierung durchgeführt.

## Phase der Migration und Modernisierung

Die Phase der Migration und Modernisierung bezieht sich auf jede Anwendung und umfasst mehrere Aufgaben, darunter die Zuweisung von Mitarbeitern, die Durchführung einer gründlichen Untersuchung, die Suche nach der richtigen Anwendungsarchitektur AWS, die Einrichtung von Laufzeitumgebungen für Anwendungen, die Neugestaltung oder Umgestaltung Ihres Codes, die Integration mit anderen Systemen und natürlich das Testen. Am Ende der Phase stellen Sie die umplattformierten oder umgestalteten Anwendungen für die Produktion bereit und wechseln dann zum neuen System auf AWS. Die meisten oder alle dieser Aufgaben werden im Rahmen der AWS Mainframe-Modernisierung, in einem anderen AWS Service oder in einem Tool ausgeführt, auf das die Mainframe-Modernisierung Zugriff bietet. AWS

[Wenn Sie automatisiertes Refactoring verwenden möchten, besuchen Sie Blu Insights.](#) AWS

Blu Insights ist jetzt AWS Management Console über Single Sign-On verfügbar. Sie müssen keine separaten AWS Blu Insights-Anmeldeinformationen mehr verwalten. Sie können sowohl auf die Funktionen AWS AWS Blu Age Codebase als auch auf das AWS Management Console Transformation Center direkt über zugreifen.

Für die Migration von Daten vom Mainframe auf empfehlen wir die AWS und die AWS SCT . AWS Database Migration Service Weitere Informationen finden Sie unter [Was ist das AWS Schema Conversion Tool?](#) im Benutzerhandbuch zum AWS Schema Conversion Tool und [Was ist AWS Database Migration Service?](#) im AWS Database Migration Service Benutzerhandbuch.

## Phase „Betrieb und Optimierung“

In der Phase Betrieb und Optimierung konzentrieren Sie sich darauf, Ihre bereitgestellten Anwendungen zu überwachen, Ressourcen zu verwalten und sicherzustellen, dass Sicherheit und Compliance auf dem neuesten Stand sind. Sie bewerten auch Möglichkeiten zur Optimierung der migrierten Workloads.

# Konzepte

AWS Mainframe-Modernisierung bietet Tools und Ressourcen, die Sie bei der Migration, Modernisierung und Ausführung von Mainframe-Workloads unterstützen. AWS

Themen

- [Anwendung](#)
- [Definition der Anwendung](#)
- [Batch-Auftrag](#)
- [Konfiguration](#)
- [Dataset](#)
- [Umgebung](#)
- [Mainframe-Modernisierung](#)
- [Reise zur Migration](#)
- [Befestigungspunkt](#)
- [Automatisiertes Refactoring](#)
- [Plattformwechsel](#)
- [Ressource](#)
- [Runtime-Engine](#)

## Anwendung

Ein laufender Mainframe-Workload im Rahmen der Mainframe-Modernisierung. AWS Eine Anwendung besteht aus einer Reihe von Batch-Jobs, interaktiven Transaktionen (CICS oder IMS) oder anderen Komponenten. Sie definieren den Umfang. Sie müssen alle Komponenten oder Ressourcen definieren und angeben, die der Workload benötigt, z. B. CICS-Transaktionen oder Batch-Jobs.

## Definition der Anwendung

Die Definition oder Spezifikation der Komponenten und Ressourcen, die für eine Anwendung (Mainframe-Workload) benötigt werden, die im Rahmen der AWS Mainframe-Modernisierung ausgeführt wird. Es ist wichtig, die Definition von der Anwendung selbst zu trennen, da es möglich ist,



dieselbe Definition für mehrere Phasen (Vorproduktion, Produktion) wiederzuverwenden, die durch unterschiedliche Laufzeitumgebungen repräsentiert werden.

## Batch-Auftrag

Ein geplantes Programm, das so konfiguriert ist, dass es ohne Benutzereingriff ausgeführt werden kann. Bei der AWS Mainframe-Modernisierung müssen Sie sowohl Batch-Job-JCL-Dateien als auch Batch-Job-Binärdateien in einem Amazon S3 S3-Bucket speichern und den Speicherort beider Dateien in der Anwendungsdefinitionsdatei angeben. Wenn Sie einen Batch-Job ausführen, meldet AWS Mainframe Modernization die folgenden Statuswerte:

### Einreichen

Der Batch-Job wird gerade eingereicht.

### Wartestatus

Der Batch-Job ist in der Warteschleife.

### Versand

Der Batch-Job wird gerade versendet.

### In Ausführung

Der Batch-Job wird derzeit ausgeführt.

### Abbrechen

Der Batch-Job wird gerade abgebrochen.

### Abgebrochen

Der Batch-Job wurde abgebrochen.

### Erfolgreich

Die Ausführung des Batch-Jobs wurde erfolgreich abgeschlossen.

### Fehlgeschlagen

Der Batch-Job ist fehlgeschlagen.

### Erfolgreich mit Warnung

Die Ausführung des Batch-Jobs wurde erfolgreich abgeschlossen und es wurde ein kleinerer Fehler gemeldet. Der als Teil der GetBatchJobExecution Antwort zurückgegebene Job-Zustandscode gibt die Ursache des Fehlers an.

# Konfiguration

Die Eigenschaften einer Umgebung oder Anwendung. Umgebungskonfigurationen bestehen aus Engine-Typ, Engine-Version, Verfügbarkeitsmustern, optionalen Dateisystemkonfigurationen und mehr.

Anwendungskonfigurationen können statisch oder dynamisch sein. Statische Konfigurationen ändern sich nur, wenn Sie eine Anwendung aktualisieren, indem Sie eine neue Version bereitstellen. Dynamische Konfigurationen, bei denen es sich in der Regel um eine betriebliche Aktivität wie das Ein- oder Ausschalten der Ablaufverfolgung handelt, ändern sich, sobald Sie sie aktualisieren.

# Dataset

Eine Datei mit Daten zur Verwendung durch Anwendungen.

# Umgebung

Eine benannte Kombination aus AWS Rechenressourcen, einer Runtime-Engine und Konfigurationsdetails, die zum Hosten einer oder mehrerer Anwendungen erstellt wurde.

# Mainframe-Modernisierung

Der Prozess der Migration von Anwendungen von einer älteren Mainframe-Umgebung zu AWS

# Reise zur Migration

Der end-to-end Prozess der Migration und Modernisierung von Legacy-Anwendungen, der in der Regel aus den folgenden Phasen besteht: Bewertung, Mobilisierung, Migration und Modernisierung sowie Betrieb und Optimierung.

# Befestigungspunkt

Ein Verzeichnis in einem Dateisystem, das Zugriff auf die in diesem System gespeicherten Dateien ermöglicht.

## Automatisiertes Refactoring

Der Prozess der Modernisierung älterer Anwendungsartefakte für den Betrieb in einer modernen Cloud-Umgebung. Er kann Code- und Datenkonvertierung beinhalten. Weitere Informationen finden Sie unter [AWS Mainframe Modernization Automated Refactor](#).

## Plattformwechsel

Der Prozess des Verschiebens einer Anwendung und Anwendungsartefakte von einer Computerplattform auf eine andere Computerplattform. Weitere Informationen finden Sie unter [AWS Mainframe Modernization Replatform](#)

## Ressource

Eine physische oder virtuelle Komponente innerhalb eines Computersystems.

## Runtime-Engine

Software, die die Ausführung einer Anwendung erleichtert.

# Automatisches Refactoring von Anwendungen mit AWS Blu Age

Das automatisierte Refactoring mit AWS Blu Age bietet eine end-to-end Lösung für die Migration und Modernisierung Ihrer Mainframe-Anwendungen. Der Refactoring-Prozess umfasst die folgenden Schritte:

- Inventar analysieren
- Analysieren Sie Abhängigkeiten
- Automatisches Transformieren von Code
- Erfassen und verwalten Sie Testszenarien

Sie können die vorherigen Schritte im Blu Insights-Tool ausführen, das über Single Sign-On in der AWS Mainframe Modernization Console verfügbar ist. Weitere Informationen zu Blu Insights finden Sie in der [Blu Insights-Dokumentation](#).

Wenn Sie mit dem transformierten Quellcode zufrieden sind, ist es an der Zeit AWS, zu wechseln, wo Sie die folgenden Schritte ausführen werden:

- Erstellen Sie die umgestaltete Anwendung und stellen Sie sie bereit.
- Stellen Sie Ihre Anwendung im Rahmen von Mainframe Modernization bereit und überwachen Sie sie. AWS

AWS Blu Age Runtime (nicht verwaltet) ist zusammen mit AWS Blu Age Managed eines der Angebote des AWS Mainframe-Modernisierungsdienstes. Mit AWS Blu Age Managed können Sie Ihre modernisierte Anwendung in einer AWS verwalteten Umgebung bereitstellen, die Ihre Benutzererfahrung vereinfacht, sodass Sie die zugrunde liegende Infrastruktur, auf der Ihre modernisierte Anwendung ausgeführt wird, nicht verwalten müssen. Im Gegensatz dazu können Sie mit AWS Blu Age Runtime (nicht verwaltet) Ihre modernisierte Anwendung in Ihrem eigenen AWS Konto bereitstellen, sodass Sie Ihre eigene Infrastruktur verwalten können. Mit AWS Blu Age Runtime (nicht verwaltet) haben Sie die Flexibilität, alle technischen Komponenten zu bedienen, die erforderlich sind, um Ihre modernisierte Anwendung so auszuführen, wie Sie es möchten.

AWS Blu Age Runtime (nicht verwaltet) ist für den Einsatz verfügbar auf:

- Amazon EC2

- Amazon ECS auf Amazon EC2
- Amazon EKS auf Amazon EC2
- Amazon ECS, verwaltet von AWS Fargate

Die Bereitstellung auf Amazon EC2 (die ersten drei Optionen in der vorherigen Liste) kann direkt in der Instance oder über eine Docker-Container-Anwendung erfolgen, was bei der Verwendung von Amazon ECS oder Amazon EKS der bevorzugte Weg ist.

#### Themen

- [AWS Versionshinweise zu Blu Age](#)
- [AWS Anweisungen zum Upgrade auf Blu Age](#)
- [AWS Blu Age Runtime-Konzepte](#)
- [AWS Blu Age Runtime Konfiguration und Konfigurationsdateien](#)
- [AWS Blu-Age-Laufzeit-APIs](#)
- [AWS Einrichtung von Blu Age Runtime \(nicht verwaltet\)](#)
- [Ändern Sie den Quellcode mit Blu Age Developer IDE](#)

## AWS Versionshinweise zu Blu Age

Dieser Abschnitt enthält die Versionshinweise der AWS Blu Age Runtime and Modernization Tools ab Version 3.5.0, die neuesten zuerst, geordnet nach Versionsnummer.

#### Note

Für Versionshinweise aus der Zeit vor diesem Dokument wenden Sie sich an den AWS Blu Age-Lieferservice. Informationen zu den neuesten Funktionen von Blu Insights finden Sie in den [Veröffentlichungen von Blu Insights](#).

#### Themen

- [Versionshinweise 4.1.0](#)
- [Runtime-Version 4.1.0](#)
- [Version 4.1.0 der Modernisierungstools](#)
- [Versionshinweise 4.0.0](#)

- [Runtime-Version 4.0.0](#)
- [Version 4.0.0 der Modernisierungstools](#)
- [Versionshinweise 3.10.0](#)
- [Runtime-Version 3.10.0](#)
- [Version 3.10.0 der Modernisierungstools](#)
- [Versionshinweise 3.9.0](#)
- [Runtime-Version 3.9.0](#)
- [Version 3.9.0 der Modernisierungstools](#)
- [Versionshinweise 3.8.0](#)
- [Runtime-Version 3.8.0](#)
- [Version 3.8.0 der Modernisierungstools](#)
- [Versionshinweise 3.7.0](#)
- [Runtime-Version 3.7.0](#)
- [Version 3.7.0 der Modernisierungstools](#)
- [Versionshinweise 3.6.0](#)
- [Runtime-Version 3.6.0](#)
- [Version 3.6.0 der Modernisierungstools](#)
- [Versionshinweise 3.5.0](#)
- [Runtime-Version 3.5.0](#)
- [Version 3.5.0 der Modernisierungstools](#)

## Versionshinweise 4.1.0

Diese Version der AWS Blu Age Runtime and Modernization Tools konzentriert sich auf Leistung und Sicherheit. Einige wichtige Funktionen und Änderungen in dieser Version sind:

- Transformation und Leistung: Damit Projekte mit einer großen Codebasis (+50 Millionen Codezeilen) erfolgreich transformiert werden können, haben wir die Leistung und den Speicherbedarf des gesamten Transformationsmechanismus optimiert.
- BAC/JAC: Sicherheit hat höchste Priorität. AWS Mit AWS Blu Age modernisierte Anwendungen müssen den Sicherheitsstandards entsprechen. Wir haben einige wichtige Upgrades an der BluSam Administration Console (BAC) und der JICS Administration Console (JAC) vorgenommen, um sie sicherer zu machen:

- Die Anwendung wurde auf Angular v17 aktualisiert.
- Zusätzlich zum nativen Support für AWS Cognito haben wir generischen Support für OAuth hinzugefügt, der Kunden mehr Flexibilität bietet, damit sie den Identitätsanbieter ihrer Wahl verwenden können.
- Die Sicherheitsfunktionen wurden mithilfe geeigneter Header konfiguriert und erweitert.
- AS400 — Unterstützung mehrerer Knoten für Datenbanksperrmechanismen. Es wurde die Möglichkeit geboten, einen gemeinsam genutzten und externen Caching-Server (Redis) miteinander zu verbinden, um eine Batch-Anwendung auf mehreren Instanzen auszuführen, z. B. die verwaltete Mainframe-Modernisierung. AWS

Diese Version der Blu Age-Laufzeit wurde mit dem folgenden Stack getestet. Andere Versionen sind möglicherweise ebenfalls kompatibel.

Komponente	Version getestet
Java	Java 17
Präsentationsebene	Knoten JS 18.18, Npm 9.8, Angular 16.1
Serviceschicht	Spring Boot 3.2.5, Spring Core 6.1.5, Spring Statemachine 4.0.0
Persistenzschicht	PostgreSQL 14, Oracle 21c
Anwendungsserver	Apache Tomcat 10.1.17

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 4.1.0

### zOS

#### Neue Features

- Konfiguration für die dynamische Handhabung von OAuth2-Anbietern hinzugefügt. SECRET\_OAUTH2\_PROVIDER\_NAME\_KEY zur Angabe des Anbieters eingeführt. Die Methode

zum Abrufen geheimer Daten wurde aktualisiert, sodass sie mehrere Anbieter verarbeiten kann. Sichergestellt, dass Geheimnisse sicher abgerufen werden von AWS Secrets Manager.

- Unterstützung für DB2-SSL-Eigenschaften hinzugefügt, sodass AWS Secrets Manager Sie ein SSL-Zertifikat (SSL TrustStore Location) und ein Passwort (SSL TrustStore Password) definieren können, um die Keystore-Datei zu entsperren.
- Unterstützung für externe Geschäftsdatenquellen hinzugefügt.
- JCL — Unterstützung für den Checkpoint-Mechanismus für den Batch-Neustart hinzugefügt.
- JCL — Unterstützung für die DCB-Parameter Record Size und RDW hinzugefügt.
- JCL — Dynamische Ordnernamenkonfiguration für generierte temporäre Dateien hinzugefügt.
- REDIS — Pool-Konfiguration zur Redis-Konfiguration für JICS hinzugefügt.
- REDIS — Datenbankindex in der Redis-Konfiguration für Catalog und JICS hinzugefügt.
- BatchScript - Die Weitergabe des Schrittnamens für laufende Programmausführungen wurde hinzugefügt.
- CICS — Unterstützung für den Befehl ADDRESS SET hinzugefügt.
- CICS — Unterstützung für PURGE MESSAGE und JUSTIFY hinzugefügt.

## Verbesserungen

- JCL - INFUTILB - Verbesserte Unterstützung für die Deaktivierung des Nullindikators basierend auf der YML-Eigenschaft.
- JCL - INFUTILB - Verbesserte Unterstützung für den CHAR/BPCHAR-Datentyp.
- JCL - ICEGENER - Unterstützung für das Kopieren mehrzeiliger Eingabestreams in Dateien hinzugefügt.
- JCL - IEBGENER - Verbesserte Unterstützung für die Konvertierung von Dateien mit variablem Block in Dateien mit festem Block.
- JCL - DFSORT - Verbesserte Unterstützung für mehrstellige Parameter bei Operation DATE.
- JCL - DFSORT - Unterstützung für die INCLUDE=ALL-Klausel hinzugefügt.
- JCL — Verbesserte Unterstützung für das SORT-Hilfsprogramm zur Verarbeitung des BDW-Felds in der Ausgabe.
- JCL — Verbesserte Unterstützung für DD-Verkettung.
- JCL — Verbesserte Unterstützung für Input Stream.
- JCL - DSNUTILB - Verbesserte Unterstützung für die NULLIF () -Anweisung.



- JCL - INFUTILB - Unterstützung für das Entladen von Daten mit der Option NOPAD hinzugefügt.
- JCL - INFUTILB - Verbesserte Unterstützung für das aktuelle Datum in INFUTILB.
- JCL - Es wurden Prüfungen auf Existenz und Größe von Dateien hinzugefügt, bevor eine Datei verwendet wird.
- JCL - GDG - Die Handhabung von Unterverzeichnissen für GDG wurde verbessert.
- MQ — Verbesserte Verbindungsöffnung in der JMS-Implementierung.
- MQ - Verbesserte Datenlängeneinstellung der GET-Nachricht für XA-Datenquelle.
- MQ — Zerlegtes CMQV-Standard-Copybook zur Vermeidung von Kompilierungsfehlern und Refactoring-Anwendungen.
- BluSam - Verbesserte Unterstützung für Löschanfragen für nicht existierende Datensätze.
- Verbesserte Unterstützung für die ALLOCATE-Anweisung.
- Verbesserte Robustheit der TS-QUEUE-Benennung.
- BatchScript - Verbesserte Beibehaltung des Rückgabecodes des vorherigen Schritts bei der erneuten Ausführung von Jobs.
- Datensatz — Die Überprüfung der Existenz von Dateien, wenn eine Datei existiert und temporär ist, wurde verbessert.
- Datensatz — Die Parallelität beim Auffinden von GDG-Dateien, die gelöscht werden sollen, wurde verbessert.
- Datensatz — Unterstützung für das Abrufen der GDG-Datensatzgröße hinzugefügt.
- CICS — Verbesserte Unterstützung für die Option SUSPENDED im Befehl INQUIRE TASK LIST.
- CICS — Verbesserte Unterstützung für LOAD SET mithilfe der ADDRESS OF-Anweisung.
- CICS — Das REMOTESYSTEM mit unbehandelten CICS-Argumenten bei CICS INQUIRE wurde verbessert.
- CICS — Verbesserte Unterstützung für den Befehl GETMAIN zur Behandlung der SET-Option mit einem mit dem Schlüsselwort OF definierten Zeiger.
- JICS — Verbesserte Robustheit der Methode jicsXAPRepare () durch Hinzufügen der Transaktionsstatusprüfung.
- JICS XA — Es wurde eine Überprüfung des Transaktionsstatus hinzugefügt und die Beendigung des Transaktionsthreads wurde verbessert.
- BAC — Verbesserte rollenbasierte Authentifizierung auf der Client-Seite und Umgestaltung/ Zentralisierung aller API-Aufrufe.

- BAC — Es wurde eine Funktion implementiert, um den öffentlichen Zugriff auf BAC und JAC je nach Konfiguration zu blockieren
- BAC — Upgrade der Abhängigkeiten: Angular 17.
- BAC - Verbesserte Sicherheitsintegration mit OAuth2 - /FIDIS. StateFarm
- BAC — Erweitertes, im Ruhezustand generiertes DDL.
- BAC — Verbesserter Mechanismus zum Exportieren von Datensätzen.
- JAC - Auf Angular 17 aktualisiert und berichtet, dass alle Einzelheiten von BAC aus funktionieren (ROLE, sadmin conf, XSRF, Logout).
- COBOL — Unterstützung für die Funktionen CHAR und ORD-MIN hinzugefügt.
- Verbessert FileFactory , um die Größe der Katalogdatensätze in der MOD-Disposition beizubehalten.
- Die Protokollierung mit MDC für JICS-Transaktionen wurde aktiviert.
- Verbessertes SQLCA > SQLSTATE, das für gespeicherte Prozeduren erstellt wurde, die Ad-hoc-Ergebnismengen generieren.
- Verbesserte Unterstützung für die Aufgabenplanung im Zusammenhang mit dem letzten Frühjahrs-Upgrade.

## AS400

### Neue Features

- Unterstützung für mehrere Knoten für Sperren von Datenbankeinträgen mithilfe von Redis hinzugefügt.
- Unterstützung für BINARY CHARACTER für den Typ DDS hinzugefügt.
- CL — Unterstützung für die Generierung von benutzerdefinierten Berichtsdateien hinzugefügt.
- RPG — Unterstützung für das Schlüsselwort RENAME für primäre/sekundäre Dateien hinzugefügt.

### Verbesserungen

- Verbesserte Datenbankunterstützung für die Behandlung der CTID-Spalte mit einer JOIN-Klausel.
- Verbesserte Cursorposition für mehrere DSPATR (PC).
- Die Protokollierung bei Leseausnahme wurde verbessert.
- Die Protokollierung von Quartz-Jobs wurde verbessert und schließt nun Jobeigenschaften in MDC ein.

- Verbesserte Unterstützung für den AS400-Hilfebildschirm.
- CL — Verbesserte Unterstützung für den Befehl RMVJOBSCDE zur Annahme von Eintragsnummern mit abschließenden Leerzeichen.
- CL — Verbesserte Unterstützung für den Befehl RMVJOBSCDE zum Entfernen eines Jobplans unter Verwendung eines generischen Jobnamens.
- CL — Verbesserte Unterstützung für den Befehl SAVOBJ zum Sortieren von Datensätzen nach Tabellenschlüsseln.
- CL — Verbesserte Unterstützung für den CPYF-Befehl zum Herstellen einer neuen Verbindung für DB-Abfragen.
- CL — Verbessertes Einfügen von Anfragenachrichten in die Warteschlangennachricht mit SNDPGMMSG.
- CL — Verbesserte Konfiguration der Jobwarteschlange zur Angabe der Standard-Jobwarteschlange.
- CL — Der CRTPF-Befehl wurde verbessert, sodass er die QTEMP-Bibliothek und den RCDLEN-Parameter unterstützt.
- CL — Verbesserte Unterstützung für den Befehl CHKOBJ — Suchen Sie nach einer Partition mit Bibliothek.
- CL — RTVMGS wurde verbessert, um CPF2407 und CPF2419 zu senden, wenn Datei/ID nicht gefunden wurde.
- CL — Verbesserte Interpretation älterer Formatierungsparameter durch CPYTOIMPF und CPYFRMIMPF.
- CL — Unterstützung für den OVRPRTF-Parameter USRDTA hinzugefügt.
- CL — Der CL-Befehl CPYTOIMPF wurde verbessert, um eine neue Verbindung herzustellen und das Schließen vorhandener Ergebnismengen zu verhindern.
- CL — CHGDTAARA wurde dahingehend verbessert, dass die Länge des Datenbereichs nicht mehr geändert wird, wenn der Inhalt aktualisiert wird.
- CL — Verbesserte CCommand Handhabung von Datenbankverbindungen.
- Optimierte Interaktion zwischen dem Frontend und dem Backend.
- COBOL — Die Transformation wurde aktualisiert, um FILLER in Copybooks zu handhaben.
- Die Anzeige zusätzlicher Nachrichteninformationen für benutzerdefinierte Nachrichten, die an das Frontend gesendet werden, wurde verbessert.
- Der Standardwert für den Selektor in app.component.ts wurde aktualisiert.

- Die Textaufteilung in der Anzeige wurde verbessert. split-dynamic-field
- Die Anzeige von Fehlermeldungen mit mehreren Schreibvorgängen, gefolgt von einem Lesevorgang, wurde verbessert.

## Transversale Fähigkeiten

### Neue Features

Unterstützung für die dynamische Konfiguration von OAuth2 Provider Secret wurde hinzugefügt.

### Verbesserungen

- Drucken — Verbesserte QCMDEXC-Parameterunterstützung für die Verarbeitung von Anführungszeichen und verbesserte Bildung von Berichtsnamen
- Verbesserte Unterstützung für Syntax mit Trennzeichen aktiviert. RecordAdaptable
- Verbesserte InspectBuilder Fehlerprotokollierung, um Kontext zur Quellzeichenfolge hinzuzufügen.
- DataSimplifier - Robustheit gegenüber Beeinträchtigungen hinzugefügt. ByteArray
- Verbesserte MDC-Protokollierung mit neuen Laufzeitattributen.

## Version 4.1.0 der Modernisierungstools

### zOS

#### Neue Features

- Unterstützung für mehrere CSD-Dateiumwandlungen hinzugefügt
- COBOL — Unterstützung für die CICS ALLOCATE-Anweisung hinzugefügt.
- COBOL — Unterstützung für ON SIZE ERROR in der Anweisung ADD CORRESPONDING hinzugefügt.
- COBOL — Unterstützung für EXIT PARAGRAPH hinzugefügt.

#### Verbesserungen

- COBOL — Verbesserte Unterstützung für -INC Copybook.
- COBOL — Verbesserte Unterstützung für die FILLER-Initialisierung.
- COBOL — Verbesserte Unterstützung für den Vergleich von Bildwerten.

- COBOL — Verbesserte Unterstützung für WHEN ANY in aufeinanderfolgenden WHEN-Klauseln ohne zwischengeschaltete Codeblöcke.
- COBOL — Verbesserte Unterstützung für figurative Konstanten.
- COBOL — Verbesserte Unterstützung für die Berechnung der Größe gepackter Typen.
- COBOL — Das unbehandelte CICS-Argument KEEP für SPOOLCLOSE wurde verbessert.
- COBOL — Verbesserte Generierung für die TEST-NUMVAL-Funktion.
- COBOL — Verbesserte Argumente für die Java-Generierung bei der Unterstützung des INSPECT-Frameworks.
- CICS — Verbesserte Unterstützung für die Definition von DFHCOMMAREA.

## AS400

### Neue Features

- RPG — Es wurde ein Mechanismus zur Fehlersuche hinzugefügt, um das (unvollständige) DDS zu generieren, sodass die Programmgenerierung nicht blockiert wird.
- Unterstützung für das Schlüsselwort INCLUDE zur Dateibeschreibung hinzugefügt.

### Verbesserungen

- RPG — Verbessertes, vollständig kostenloses Parsen.
- RPG — Robustheit beim Abfangen von Fehlern wurde hinzugefügt.
- RPG — Verbesserte Initialisierung von Feld/DS mit Export-Schlüsselwort.
- RPG — Verbesserte DAO-Bedienung zur Handhabung von Indikatoren.
- RPG - Hat den Standardwert von PERRCD mit CTDATA behandelt.
- RPG — Der Free-RPG-Parser wurde aktualisiert, sodass pro Parserregel ein eindeutiger Fehler protokolliert wird.
- PRTF — Die Namenskollision zwischen PRTF und JRXML wurde behoben.
- COBOL — Verbesserte Unterstützung für das Schlüsselwort LIKE.

## Transversale Fähigkeiten

### Verbesserungen

- Robustheit für die ErrorID-API wurde hinzugefügt
- Leistungsoptimierung für die Transformation großer Projekte. Zum Beispiel: Zeitlimit für das Überspringen blockierter Dateien, Wiederverwendung der Klassifikation aus Blu Insights und bessere Speicherzuweisungen.
- Der Speicherbedarf während der COBOL/PL1-Transformation wurde optimiert.
- CVE auf Drittanbietern (jQuery und Bootstrap) wurde behoben.
- Verwaltete TimeoutParser-Optionen in TC.
- Das Umschreiben mehrerer Leerzeichen bei SQL-Abfragen wurde verbessert.
- Verbesserter Schreibgeschützter Cursor mit Sensitivitätsattribut.

## Versionshinweise 4.0.0

Anweisungen zur Migration von AWS Blu Age Runtime 3.10.0 auf 4.0.0 finden Sie unter [the section called “Migration von 3.10.0 auf 4.0.0”](#)

Diese Version von AWS Blu Age Runtime and Modernization Tools konzentriert sich auf die Aktualisierung kritischer Abhängigkeiten und unterstützter Technologien bei gleichzeitiger Steigerung der Leistung mehrerer Funktionen. Einige der wichtigsten Funktionen und Änderungen in dieser Version sind:

- Führen Sie ein Upgrade von Spring Boot 2.7 auf 3.2.4, Spring Core 5.3 auf 6.1.5 und Tomcat 9.0 auf 10.1.17 durch, um die Sicherheit, Leistung und Wartbarkeit zu verbessern, indem Sie Versionen verwenden, die aktiv gepatcht und gewartet werden.
- Lazy Loading in der Frontend-Anwendung, um größere Projekte mit mehr als 2000 Bildschirmen schneller zu erstellen und die Initialisierung der Anzeige von 10 s auf 300 ms zu reduzieren.
- Support für die DBCS-Anzeige in der Frontend-Anwendung. Verbesserung der Unterstützung von Doppelbyte-Zeichen, um eine neue Schriftart bereitzustellen, die Doppelbyte- und Einzelbyte-Zeichen verarbeitet, Einzelbyte-Eingaben in einem Doppelbyte-Feld verhindert und Felder mit gemischten Doppelbyte- und Einzelbyte-Zeichen verarbeitet.
- Thread-Überwachungsfunktion für die AS400-Online-Anwendung zur Ausführung von AS400-Anwendungen mit Parallelisierung.
- Verbesserte Leistung bei Kontext und RunUnit Initialisierung durch Hinzufügen eines konfigurierbaren Mechanismus zur Vorinitialisierung des Programmkontextes, wodurch die Auswirkungen des Ladens komplexer Strukturen reduziert werden, die mit der Komplexität älterer Systeme einhergehen.

Diese Version der AWS Blu Age Runtime wurde mit dem folgenden Stack getestet. Andere Versionen sind möglicherweise ebenfalls kompatibel.

Komponente	Version getestet
Java	Java 17
Präsentationsebene	Knoten JS 18.18
	Npm 9.8
	Eckig 16.1
Serviceschicht	Spring Boot 3.2.4
	Spring Core 6.1.5
	Spring Statemachine 4.0.0
Persistenzschicht	PostgreSQL-Engine 14
	Orakel 21
Anwendungsserver	Apache Tomcat 10.1.17

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 4.0.0

### zOS

#### Neue Features

- Unterstützung für die Include-Anweisung '-INC CPYNAME' wurde hinzugefügt.
- CICS — Unterstützung für die PUSH/POP HANDLE-Anweisung hinzugefügt.
- COBOL — Unterstützung für „ASSIGN TO DYNAMIC“ hinzugefügt.
- Unterstützung für DB2 UNLOAD mit INFUTILB hinzugefügt.

- Unterstützung für das Schlüsselwort SEQNUM in einer OVERLAY of INREC-Anweisung hinzugefügt.

## Verbesserungen

- SORT - Unterstützung für Sonderzeichen (Klammern und Sternchen) in Sortierzeichenfolgenliteralen C'... 'hinzugefügt.
- SORT — Verbesserte Unterstützung für das Argument UTFIL NOMATCH- (..).
- SORT — Unterstützung für die SYMNames-Datendefinition hinzugefügt.
- SORT — Verbesserte Behandlung der Argumente TO= und LENGTH=.
- SORT — Die Behandlung bei der MOD-Disposition wurde verbessert.
- SORT — Unterstützung für das Argument HIT=NEXT hinzugefügt.
- ICEGENER wurde verbessert, um Unterstützung für die spezifische Kodierung von Ausgabedateien hinzuzufügen.
- INFUTILB — Verbesserte Unterstützung für die WITH UR-Klausel.
- INFUTILB — Verbesserte Unterstützung für Unload, wenn der Wert falsch ist. writeNullIndicator
- DSNUTILB — Verbesserte Robustheit gegenüber Ladeschritten, wenn das NULLIF-Schlüsselwort hinter einem optionalen SQL-Schlüsselwort steht.
- DSNUTILB — Verbesserte Unterstützung für isolierte Spaltennamen.
- DSNUTILB — Unterstützung für das Laden einer leeren Datei in eine Tabelle hinzugefügt.
- DNSUTILB — Unterstützung für die MOD-Disposition für die DNSUTILB SYSDISC-Datei hinzugefügt.
- IDCAMS — Verbesserte Unterstützung für Kommentare.
- JCL — Unterstützung für Spalten mit doppelten Anführungszeichen hinzugefügt. LoadTask
- JCL — Verbesserte UNLOAD-SQL-Abfragebehandlung in Bezug auf das Entfernen von Leerzeichen.
- JCL — Verbesserte Reaktion des Groovy-Skripts, wenn bei der Verarbeitung eine Ausnahme auftritt, um ein JSON-Format sicherzustellen.
- JCL — Verbesserte Disposition der Prüfdateien im Fall von DISP=NEW und DISP=OLD.
- JCL — Verbesserte Unterstützung für die Verarbeitung mehrerer GDG-Generierungsreferenzen mit Sonderzeichen im GDG-Basisnamen.
- JCL — Verbesserte Unterstützung für das Laden einer Dummy-Datei.



- JCL — Verbesserte Unterstützung für tempFilesDirectory YML-Parameter.
- JCL — Verbesserte JSON-Rückgabe, wenn sie benötigt wird, um doppelte Anführungszeichen innerhalb eines Zeichenkettenelements zu umgehen.
- JCL — Verbessert, FileUtils um den GDG-Basisnamen zu unterstützen.
- JCL — Erweitertes DSNTEP-Programm für die Ausführung mehrerer DB2-Abfragen.
- Unterstützung für Spring Beans hinzugefügt.
- Der SQLConverter wurde verbessert, um das Korrigieren falscher Daten zu vermeiden.
- Verbesserte JicsTimeBuilder Handhabung von YYYYDDD.
- Auf benutzerdefinierte Jars konnte von Groovy aus zugegriffen werden.
- IMS — Verbesserte Navigation zwischen Datensätzen in der IMS-Datenbankimplementierung.
- IMS — CBLTDLI wurde erweitert, sodass das Programm jetzt mit Purge gestartet werden kann.
- IMS - DFSRRC00 kann die Parameter vom Groovy-Programm an das Backend-Programm übergeben.
- Unterstützung für JICS-Befehle hinzugefügt, die nicht über einen TransactionRunner aufgerufen wurden.
- JICS — Verbesserte Leistung durch Verwendung eines konfigurierbaren Caches.
- BluSam - Unterstützung für die Deaktivierung des Aufwärmens BluSam beim Öffnen hinzugefügt, um die Leistung bei großen Datensätzen zu verbessern.
- BluSam- Verbessertes Lösch-/Umbenennungsverhalten bei regulären Datensätzen. BluSam
- BluSam - Verbesserte Leistung bei Aufzeichnungsoperationen.
- Verbesserter Datenvereinfacher für Methoden, die bestimmen, ob eine Zeichenfolge einen niedrigen Wert hat.
- Verbesserte Unterstützung für Probleme mit Packed-Decimal und Sortierreihenfolge.
- Verbesserte Konfiguration von DB2 als primäre Datenquelle mit AWS Secrets.
- Verbesserte FileSystem API zur Offenlegung des Dateistatus.
- Verbesserte DynamicFileBuilder Lesestream-Eingabe mit LineSeparator.
- Verbesserter Datenverstärker für Methoden, die bestimmen, ob eine Zeichenfolge einen niedrigen Wert hat, wenn es um den CUSTOM930-Zeichensatz geht.
- SQL — Verbesserte Ausgabeverarbeitung für gespeicherte SQL-Prozeduren.
- SQL — Verbessertes Lambda-Mapping für mehrere Tabellen mit Aliasnamen.

- COBOL — Verbesserte Unterstützung für die LENGTH OF-Anweisung.
- COBOL — Unterstützung für die TRANSFORM-Anweisung hinzugefügt.
- COBOL — Unterstützung für 9 neue mathematische Funktionen hinzugefügt.
- COBOL — Verbesserte Unterstützung für INTEGER-OF-DAY-FUNKTION.
- COBOL — Verbesserte Unterstützung für 88-Level mit figurativen Werten.
- COBOL — Verbesserte Transformation für die SET ADDRESS-Anweisung.

## AS400

### Neue Features

- Doppelte Indikator-Entitäten wurden entfernt.
- Unterstützung für DBCS-Zeichen hinzugefügt.
- Die Handhabung des HELP-Schlüsselworts für die Steuerung von Unterdateidatensätzen wurde eingeführt.
- Es wurde ein Konfigurationsparameter hinzugefügt, um die Groß- und Kleinschreibung von Spaltennamen umzuschalten und den Inhalt der Kommentarspalte in einem Pipe-Zeichen aufzuteilen.
- Unterstützung für die Verwendung von 0x0c als letzten Nibble für Felder vom Typ Packed wurde hinzugefügt.
- RPG — Behandelte Prototypen, die mit ExtProc ('System') deklariert wurden.
- CL — Der behandelte 'CLEAR' Parameter des CL-Befehls RMVMSG + führt zu speicherinternen Nachrichtenwarteschlangen, die nicht zum Programm gehören.
- CL — Behandelte generische Anweisungen, die an SBMJOB-CMD () -Aufrufe übergeben wurden.
- CL — Die Befehle STRCMTCTL und ENDCMTCTL wurden hinzugefügt. Modifizierter Sperrmechanismus und Bereinigung von Transaktionen und Sperren.
- CL — Unterstützung für den Parameter RCDDLML für den Befehl CPYTOIMPF hinzugefügt.
- CL — Der Befehl SAVOBJ unterstützt jetzt das Auffüllen von Nullen.
- CL — Die Behandlung von Bibliotheken, die im qualifizierten Namen des OBJ-Parameters für RTVOBJD enthalten sind, wurde hinzugefügt.
- CL — Unterstützung für die CPYTOIMPF-Befehlsparameter STRDLM, STRESCCHR und RMVBLANK hinzugefügt.

- CL — RTVMGS wurde verbessert, sodass CPF2407 und CPF2419 gesendet werden, wenn die Datei/ID nicht gefunden wurde.
- CL - Verbesserter RCVF-Befehl zum Empfangen von Datensätzen aus jeder bereitgestellten Bibliothek im DEV-Parameter.

## Verbesserungen

- Die Standardwerte für den Blu4IV Task Executor wurden geändert, um standardmäßig eine bessere Skalierung zu ermöglichen.
- Parameterhelper wurde geändert, um die Liste von Zeichenketten und in eine Zeichenfolge zu konvertieren. ElementaryRangeReference
- CTID wurde verbessert, um nicht existierende Spalten in POSTGRE zu verarbeiten.
- Robustheit zur Unterstützung der Userspace-API „QUSPTRUS“ hinzugefügt.
- Unterstützung für die User Spaces-APIs QUSRUSAT und QUSCUSAT hinzugefügt.
- Verbesserte Unterstützung für die User Space API (QUSPTRUS) ohne Fehlercode.
- Unterstützung für CRON Job Scheduling mit Quartz hinzugefügt.
- Verbesserte Unterstützung des RPG-Programmzyklus.
- Verbessertes Blu4IV-Transaktionsmanagement.
- Das Sperren von Datensätzen von Dateien, die innerhalb derselben Transaktion unter Commitment-Kontrolle stehen, wurde verbessert.
- Die Handhabung der Initialisierung von Unterdateien wurde verbessert.
- Verbesserte Anzeige der Scroll-Indikatoren für Nachrichtenzeilen.
- Nachfolgende Nullen bei Zahlen, die über die Datenwarteschlange gesendet wurden, wurden verhindert.
- Der Bildschirm mit zusätzlichen Nachrichteninformationen wurde verbessert.
- Die JPA-Schreiboperationen wurden verbessert, um die aktuelle Bibliothek zu berücksichtigen.
- Verbessertes Verhalten ProgramJobExecutor bei der Ausführung von Programmen ohne Parameter.
- Es wurde eine Funktion hinzugefügt, um Argumente von Frontend-Links direkt an Backend-Skripte zu übergeben.
- Verbesserte Transaktionsverarbeitung für Job-Metadaten.
- CL — Unterstützung für den Parameter SECLVL in RTVMSG hinzugefügt.

- CL — Leere Implementierung für CLRLIB hinzugefügt.
- CL — Verbesserte CPYFRMIMPF-Unterstützung für das Kopieren sowohl aus der Datenbank als auch aus CSV.
- CL — Verbesserte CPYFRMIMPF-Implementierung, um zusätzliche Spalten zu ignorieren.
- CL — Verbesserte Interpretation älterer Formatierungsparameter durch CPYTOIMPF und CPYFRMIMPF.
- CL — Param hinzugefügt, um numerische Werte in SAVOBJ zu formatieren. removeDecimalPoint
- CL — Der RCVF-Befehl wurde verbessert, um die EOF-Bedingung korrekt zu behandeln.
- CL - RTVSYSVAL - Implementierung SYSVAL = QDATETIME.
- CL - Der OVRDBF-Befehl wurde geändert, um FileID als Standardtabellennamen zu erhalten.
- CL - RTVJOBA Unverfügbarer Wert für den Parameter: USRLIBL.
- CL — Führende Schrägstriche im MSGF-Parameter SNDPGMMMSG wurden verarbeitet.
- CL — Verbesserte Unterstützung für Platzhalter in der Quelldatei im Befehl DSPFFD.
- CL — Verbesserte Handhabung des Parameters PGMQ in RCVMSG und SNDPGMMMSG.
- CL — Der RTVMSG-Parameter MSG wurde optional gemacht, um ihn an ältere Dokumente anzupassen.

## Transversale Fähigkeiten

### Neue Features

- Verbesserte Fähigkeit beim Übergeben von Parametern an die USING-Klausel des OPEN-Cursors.
- Leistung: Verbesserte Vorinitialisierung des Kontextes und RunUnit zur Leistungsoptimierung.

### Verbesserungen

- Der Mechanismus zum Ausgeben niedriger Werte aus dem UNLOAD-Befehl des INFUTILB-Hilfsprogramms wurde verbessert.
- Unterstützung für die aktuelle Schemaoption im Secret Manager von DataSources hinzugefügt.
- Verbesserte Laufzeit, sodass Parameter, die beim Öffnen des Cursors übergeben werden, nicht berücksichtigt werden, wenn sie nicht benötigt werden.
- Die Validierung des numerischen Formats für numerische Felder wurde verbessert.

- Verbesserte SQL-Diagnose in einer hochparallelen Ausführungsumgebung.
- Unicode für die Codepage-Bytesequenz (FE FD) wurde eingeführt.
- DataSimplifier Leistungsoptimierung — Verbesserte Zuweisungsanweisungen.
- DataSimplifier Leistungsoptimierung — Verbessern Sie den Standardwert für die Initialisierung numerischer Typen, um eine nutzlose Verwendung zu verhindern. BigDecimal

## Version 4.0.0 der Modernisierungstools

### zOS

#### Neue Features

- Unterstützung für das Abend-Programm hinzugefügt.
- Verbesserte Unterstützung für die Generierung von AIX-Datensätzen.
- COBOL — Unterstützung für die JUSTIFIED-Klausel in alphanumerischen/ALPHABETISCHEN/GRAFISCHEN Feldern hinzugefügt.

#### Verbesserungen

- Verbesserte PURGETHRESH-Attributbehandlung für TRANSCLASS-Ressourcendefinitionen.
- Verbesserte Unterstützung für Datendefinitionen und MOVE-Anweisungen.
- CICS — Verbesserte Unterstützung für den Befehl DELAY bei der Option MILLISECS.
- Verbessertes SQL-Lambda-Mapping für mehrere Tabellen mit Aliasen.
- Verbesserte Unterstützung für die Suche nach übergeordneten Feldern.
- Verbesserter SQLCA-Sqlstate-Satz für COMMIT- und ROLLBACK-Operationen.
- COBOL — Verbessern Sie das Parsen, indem Sie veraltete Absätze kommentieren
- COBOL — Verbesserte Unterstützung für die REPLACING-Klausel.
- COBOL — Unterstützung für mathematische Funktionen ASIN ACOS LOG TAN hinzugefügt.
- COBOL — Unterstützung für mehrere AFTER-Anweisungen in PERFORM VARYING hinzugefügt.
- COBOL — Verbesserte Unterstützung für RAMES-Felder (Stufe 66).
- COBOL — Verbesserte LENGTH OF-Methode zum Abrufen der Länge an einem bestimmten Index in einem Array-Feld.

- COBOL — Unterstützung für mehrere AFTER-Klauseln in PERFORM VARYING-Anweisungen hinzugefügt.
- COBOL — Verbesserte Unterstützung für die RAMES-Klausel.
- COBOL — Verbesserte Unterstützung des Schlüsselworts PICTURE.
- COBOL — Verbesserte Unterstützung für Level-88-Feldanalyse.
- COBOL — Verbesserte Zugriffsqualität bei Tabellendatenelementen.

## AS400

### Neue Features

- Funktionalität zur Übergabe von Argumenten an direkte Frontend-Java-Aufrufe hinzugefügt.
- CL — Verbesserte %SST-Generierung, einschließlich Unterstützung für\*LDA mit CL→Java.
- RPG — Es wurde ein vom Programm beschriebener Datensatz für DISK-Dateien hinzugefügt.

### Verbesserungen

- Verbesserte Anzeigedatei, Auflösung referenzierter Felder mit dem Schlüsselwort „REFFLD“.
- Verbesserte Unterstützung für das Schlüsselwort SETOF-CSRLOC für die Anzeigedatei.
- Dateien wurden nach dem Schließen aus der Commitment-Kontrolle entfernt.
- Es wurde ein konsistentes Verhalten bei gleichzeitigen Lese- und Schreibvorgängen an einer Tabelle gewährleistet, wenn sie von demselben Programm ausgeführt wurden.
- Die Zuweisung zur Teilzeichenfolge von wurde verarbeitet. SizePrefixedAlphanumericType
- Hat die Übergabe der Datenstruktur an eine Prozedur mit einem Zeichenkettenparameter unterschiedlicher Länge behandelt.
- Die Beibehaltung ungültiger numerischer Werte beim OnBlur-Ereignis und die Erstellung von Ereignis-Listenern nur für gültige Felder wurden verbessert.
- Die Fehlermeldungen auf Bildschirmen und die Markierung von Feldern mit ungültiger Eingabe wurden verbessert.
- Verbesserte Handhabung von Bildschirmfeldern, die von Indikatoren abhängig sind.
- Scrollen mit dem Mousrad aktiviert.
- Unterstützung für Funktionstasten für den Helpscreen hinzugefügt.

- Verbesserte Unterstützung für langen Text in split-dynamic-field Komponenten.
- Verbesserte Handhabung von LF-Dateien mit mehreren Datensätzen beim Umbenennen von Datensätzen.
- CL - Verbesserter RTVJOBDB-Befehl zur Verarbeitung von LF-Dateien (Ansichten).
- CL — Verbesserter OVRDBF-Befehl bei Verwendung auf einem LF mit mehreren Datensätzen.
- RPG — Behandeltes Szenario, in dem die Prozedur eine Variable mit demselben Namen wie der umbenannte Parameter definiert.
- RPG — Verbesserte Handhabung von \*ZEROS bei der Initialisierung von signiertem BinaryInteger.
- RPG — Verbesserte Behandlung von Zeigern auf nicht-lokale (Referenz-) Variablen.
- RPG — Verbesserte Behandlung von ELSEIF-Anweisungen, die auf iFXX-Anweisungen folgen.
- RPG — Unterstützung für Felder hinzugefügt, die im Prototyp mit LIKE definiert wurden.
- RPG — Die Unterstützung für das Schlüsselwort LIKE in einem von LIKEREK erstellten Feld wurde verbessert.
- RPG — Verbesserte Generierung von Operatoren mit figurativen Elementen.
- RPG — Die Analyse des Array-Ausdrucks xxx (\ \*) wurde verbessert und unterstützt ihn in %lookup.
- RPG — Verbesserter LookUp Operationscode mit hohen und gleichen (oder niedrigen und gleichen) Indikatoren.
- RPG — Verbessertes Freiform-Parsing.
- RPG — Verbessertes Parsen von I-Card-benannten Konstanten, die den I-Card-Datensatzformaten folgen.
- RPG — Verbesserte Unterstützung für die Typen INTEGER und UNSIGNED.
- COBOL — Unterstützung für INDIC-Klausel im DSPF-Format in der COPY DDS-Anweisung hinzugefügt.
- COBOL — Verbesserte Grammatik für DISPLAY- und ACCEPT-Anweisungen, um Transformation und Generierung zu entsperren.
- COBOL — Unterstützung für DISK-Dateien hinzugefügt.
- COBOL — Verbesserte Programme zur Unterstützung von DDS-Anzeigedateien.
- COBOL — Unterstützung für die LIKE-Klausel hinzugefügt.
- COBOL — Unterstützung für die vom Programm beschriebene DISK-Datei hinzugefügt.

- COBOL — Unterstützung für Dateinamen mit Suffix hinzugefügt.

## Transversale Fähigkeiten

### Neue Features

- Hat das verzögerte Laden von Kartenkomponenten von Webprojekten übernommen.

### Verbesserungen

- Verbesserte Java-Generierung von SQL-Indikatorparametern.
- Verbesserte Fähigkeit, Variablen zu verarbeiten, die an der SET DB2-Anweisung beteiligt sind.
- Verbesserte Fehlerauslösung am Ende des abgerufenen Cursors, wenn es sich bei der Ausgabe um ein einzelnes Entity-Array handelt.
- Verwalteter Pfad unter Linux.
- Data Migrator verwaltet Sicherheitslücken und entfernt ungenutzte Abhängigkeiten.

## Versionshinweise 3.10.0

Diese Version von AWS Blu Age Runtime and Modernization Tools konzentriert sich auf grundlegende Upgrades und Verbesserungen des gesamten Produkts, um die Leistung und Robustheit in allen Transformations- und Ausführungsschritten zu erhöhen. Einige der wichtigsten Funktionen und Änderungen in dieser Version sind:

- Versionsupgrade von Java 8 auf Java 17, was die Sicherheit und Leistung erhöht und es Kunden ermöglicht, in einer moderneren Sprache implementierte Anwendungen bereitzustellen und auszuführen und aktuelle Framework-Versionen von Drittanbietern zu verwenden.
- Zusätzliche Unterstützung für die Verwaltung großer gemeinsam genutzter Speicherbereiche zwischen Benutzern oder Jobs sowie für das Speichern von Daten, die nach dem Neustart der Anwendung oder Instanz wiederverwendet werden können.
- Schnellerer Zugriff auf große Datensätze in Blusam mithilfe eines Paginierungsmechanismus, der es ermöglicht, eine Teilmenge von Datensätzen inkrementell abzurufen.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.



## Runtime-Version 3.10.0

Diese Runtime basiert auf Java17, Spring2.7 und Angular16.

### zOS

#### Neue Features

- Blusam — Unterstützung für große Datensätze wurde durch einen Mechanismus mit Seitennummerierung hinzugefügt, bei dem Indizes mithilfe von Seiten gespeichert und geladen werden

#### Verbesserungen

- DataUtils.compare wurde verbessert, um die Konvertierung von einer Zeichenfolge in eine Zahl mit niedrigerer Priorität zu handhaben
- Mithilfe der YML-Eigenschaft DataSimplifier.Byte Check kann ByteRange nun überprüft werden, ob kein Wert mit falschen Werten erstellt wurde RangeBounds
- removeSOSI () wurde verbessert, um die Initialisierung von a mit einem leeren Zeichen zu unterstützen GraphicAlphanumericType
- Robustheit für den Jobbetrieb und das sichere Lesen des GDG-Status wurde hinzugefügt
- Blusam — Unterstützung für das Löschen des Ehcache von Blusam-Datensätzen mithilfe einer neuen Methode namens .removeCache () wurde hinzugefügt CoreBluesamManager
- Blusam — Verbessertes Lösch-/Umbenennungsverhalten für reguläre Blusam-Datensätze
- Redis — Verbesserte Unterstützung für das Entsperren von Datensätzen und das Aufheben der Datensatzsperr
- JICS — Die Fehlermeldung für fehlgeschlagene Anfragen wurde verbessert
- JCL — Unterstützung für die Verkettung von ControlM-Variablen auf der Grundlage von Punktzeichen hinzugefügt
- JCL — Unterstützung für Write ADVANCING (ADV) für GDG-Dateien hinzugefügt
- JCL — Verbesserte Unterstützung für die aktuelle Generationsnummer nach dem Löschen aller GDG-Dateien
- JCL — Verbesserte Unterstützung für das Lesen von RDW/RecordSize aus dem Katalog bei der Datensatzerstellung

- JCL — Es wurde Unterstützung hinzugefügt, um das Ressourcenobjekt (von AbstractSequentialFile) beim Öffnen der Datei mit der Größe des Datenausgabedatensatzes zu aktualisieren
- JCL — Verbesserte IDCAMS-Leistung
- JCL — Verbesserte Unterstützung für PRINT STATEMENT durch Hinzufügen von „CHAR“ als Alias für „CHARACTER“
- SORT — Verbesserte Unterstützung für den Kopiervorgang von einem Blusam-Datensatz mit fester Länge in einen Datensatz mit variabler Länge
- SORT — Verbesserte Sortiergrammatik, um einige spezifische Anweisungen verarbeiten zu können

## AS400

### Neue Features

- Unterstützung für User Spaces und die zugehörigen APIs wurde hinzugefügt
- Unterstützung für den TOMSGQ-Parameter von SNDPGMMSG und implementierte Nachrichtenwarteschlangen hinzugefügt
- CL — Unterstützung für die Parameter FILE und SPLFNAME für den Befehl OVRPRTF hinzugefügt
- CL — Unterstützung für die Handhabung von Bibliotheken für entsprechende Partitionstabellen mit dem CPYF-Befehl hinzugefügt
- CL — Unterstützung für den Umgang mit dem Befehl CHGCURLIB und die Berücksichtigung der aktuellen Bibliothek bei der Erstellung von Abfragen hinzugefügt
- CL — Unterstützung für die Verarbeitung des Befehls cl als Teil des Aufrufs stacktrace hinzugefügt

### Verbesserungen

- Verbessert MessageHandlingBuilder für eine bessere Handhabung des Call-Stack-Trace-Eintrags
- Verbesserte parallel Ausführung der ContextPreConstruct-Funktion
- Verbesserte Anzeigeattribute, wenn ein Datensatz von SFLINZ erstellt wird
- SAVOBJ wurde verbessert, um die Verarbeitung mehrerer Ausgabedateien zu ermöglichen
- Die Handhabung von Groovy-Programmen wurde verbessert, indem sie hinzugefügt wurden, programCallStack wenn sie von einem Java-Programm aus aufgerufen werden

- Die Erkennung der obersten Position von Help Modal wurde verbessert
- Verbesserte TopGMQ-Funktionalität, wenn der TomSGQ-Parameter für SNDPGMMSG bereitgestellt wird
- Das Abrufen vordefinierter Nachrichten und die Funktionalität des Message Loaders wurden verbessert
- Verbesserte CPYTOIMPF-Behandlung von Trennzeichen im Inhalt
- Die Freigabesperre für den READ-Datensatz wurde verbessert

## Transversale Fähigkeiten

### Neue Features

- Es wurde eine Übersetzung für Systemnachrichten im Front-End hinzugefügt
- Es wurde eine neue Methode hinzugefügt ExecutionContext , um den Programmaufruf-Stack zurückzugeben
- Stellen Sie unabhängig von der tatsächlichen Umgebung ein Zeilentrennzeichen (zur Datenvereinfachung) ein
- Es wurde die Möglichkeit hinzugefügt, den JSON-Pfad des SQL-Modells zu konfigurieren

### Verbesserungen

- Die Vergleichsmethode DataUtils .compare AlphInt () wurde verbessert, wenn es um Padding geht
- Erstellung eines Flags, um benutzerdefiniertes Verhalten bei Ausnahmen in Cursorabfragen zu ermöglichen
- Verbesserte grafische LOWVALUES-DB-Konvertierung

### Drittanbieter

- Führen Sie ein Upgrade durch, um die Schadensbegrenzung zu verhindern (CVE-2024-21634, CVE-2023-34055, CVE-2023-34462, IN1-JAVA-ORG/SPRINGFRAMEWORKSECURITY-5905484, CVE-2023-46120, CVE-2023-6481, CVE-2023-6378, CVE-2023-5072)

## Version 3.10.0 der Modernisierungstools

### zOS

#### Verbesserungen

- COBOL - Unterstützung für die ABS-Funktion hinzugefügt
- JCL — Erweiterter Variablenbereich: an STEP statt an JOB angehängt
- Verbesserte Eingabe von Cursorparametern für niedrige/hohe Werte
- Verbessertes CSD-Parsing, insbesondere für Remote-TRANSAKTIONEN

### AS400

#### Verbesserungen

- Das leere Häkchen für die Kontrollstandsanzeige wurde entfernt
- Unterstützung für externe Namen für IMPORT/EXPORT-Schlüsselwörter hinzugefügt
- Unterstützung für %LEN in Feldern wurde hinzugefügt
- CL — Unterstützung für neue Operatoren für die CLLE-Sprache hinzugefügt
- CL — Unterstützung für verschachteltes IF hinzugefügt
- COBOL — Verbesserte Handhabung des START-Befehls bei Verwendung mit mehreren Tasten
- DSPF — Verbesserte Behandlung der Cursorposition mit der Datensatznummer
- DSPF — Die Formatierung für numerische Felder mit Vorzeichen, nur numerische Felder und Felder mit großem Maßstab wurde verbessert
- DSPF — Die Bestimmung des Titels für die allgemeine Bildschirmhilfe wurde verbessert
- DSPF — Verbesserte Unterstützung von Eingabe-/Ausgabespezifikationen
- DSPF — Verbesserte Handhabung von Gruppentrennzeichen bei der Validierung von numerischen Feldern
- Verbesserte Zuordnung von Ausgabe-/DDS-Datensätzen
- Verbesserte Fähigkeit von REFFLT-Schlüsselwörtern für Druckerdateien, referenzierte Felder aufzulösen
- RPG — Verbesserte Unterstützung für „ALLE kostenlosen“ Kontoauszüge

- RPG — Die Analyse von Zuständen wurde verbessert und es wurde Unterstützung für den Umgang mit CABXX ohne Ergebnis-TAG hinzugefügt
- RPG — Verbesserte Handhabung von Eingabespezifikationen für numerische Felder
- RPG — Verbesserte Behandlung von Prozeduraufrufen innerhalb von IF/ELSEIF/WHEN-Bedingungen
- RPG — Verbesserte Handhabung des READ-Befehls, wenn er in einer DSPF-Datei aufgerufen wird
- RPG — Verbessert die Unterstützung für Dateien, die auf ein nicht existierendes DDS verweisen
- Verbessert die Handhabung von REFFLD, wenn ein physischer Datensatzformatname übergeben wird
- Unterstützung für die Verwendung von 'return' als Datenbankspaltenname hinzugefügt

## Transversale Fähigkeiten

### Neue Features

- Oracle - Es wurde möglich, Benutzer als SYS zu definieren, um integrierte Funktionen zu speichern

### Verbesserungen

- Java-Version von v8 auf v17 aktualisiert
- Die SQL-Bedingung mit dem Cluster-Spaltennamen wurde verbessert
- Unterstützung für ORDER BY-Klauseln aus View hinzugefügt

## Versionshinweise 3.9.0

Diese Version von AWS Blu Age Runtime and Modernization Tools konzentriert sich auf mehrere bereichsübergreifende Verbesserungen im gesamten Produkt, mit denen die Leistung in Architekturen mit hoher Verfügbarkeit gesteigert werden soll, sowie auf neue Funktionen, um die Ausführung von Jobs auf die nächste Stufe zu heben. Einige der wichtigsten Funktionen und Änderungen in dieser Version sind:

- Versionsupgrade von Angular 13 auf Angular 16, wodurch die Sicherheit erhöht und der Zugriff auf neue Funktionen ermöglicht wird, die die Leistung der Online-Anwendungen der Kunden verbessern.

- Fügen Sie Unterstützung für jobübergreifende Funktionen in AS400 hinzu, wobei der Hauptvorteil darin liegt, dass Jobs Anfragenachrichten synchron zwischen ihnen senden können, was die Entkopplung bei modernisierten Jobs ermöglicht.
- Leistungsverbesserungen bei der Verwendung von Redis, einschließlich Optimierung des Verbindungspools, hoher Verbindungssicherheit und verbesserter Mechanismus zum Sperren von Datensätzen.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 3.9.0

### zOS

#### Neue Features

- Sortierprogramm: Aktualisierte VSAM-Eingänge mit fester Länge
- JHDB DB: Konfigurierbares Timeout hinzugefügt

#### Verbesserungen

- Verbesserte Unterstützung für Zeilentrennzeichen zum Streamen, wenn sie bei der Dateiverkettung verwendet werden
- Verbesserte Unterstützung für das Öffnen von verketteten sequentiellen Dateien. Nach dem Öffnen der Datei initialisieren DataSetIndex
- Verbesserte Unterstützung für virtuelle Dezimaltrennzeichen, wenn a auf einen numerischen Wert angewendet NumericEditedType wird
- Verbesserte Unterstützung für „NumericEditedType Keine negativen Werte“
- IDCAMS: SYSIN-Karten werden jetzt mit der in .yml definierten Eigenschaft „encoding“ gelesen application-utility-pgm
- IDCAMS: Die Grammatik wurde aktualisiert, um das Argument FILE (..) in der Anweisung DEFINE CLUSTER zu unterstützen
- INFUTILB: Unterstützung für das DFSIGDCB-Argument hinzugefügt, um DCB-Parameter von DD SYSREC zu überschreiben
- INFUTIL: Verbesserte Unterstützung für den Parameter „DFSIGDCB YES“

- SPLICE wurde verbessert, um große Eingabedateien zu verarbeiten
- DFSORT: Verbesserte Handhabung von Bemerkungsfeldern
- DFSORT: Unterstützung für das (signierte/unsigned) numerische Freiformformat (SFF/UFF) hinzugefügt
- SORT: Parsing-Unterstützung für die Anweisungen OPTION PRINT und OPTION ROUTE hinzugefügt
- SORT/ICEMAN: Unterstützung für geschlossene Divisionsoperationen hinzugefügt (Feld mit DIV-Operator)
- Verbesserte Unterstützung für CICS READ unter Verwendung eines generischen Schlüssels
- Die Funktion StringUtils .chgraphic wurde korrigiert, um SOSI aus einem Grafiktyp zu entfernen
- Verbessern Sie die Leistung bei .is Encoding DataUtils DoubleByte
- JCL: Verbesserte Unterstützung für den KEEP-Dispositionsmodus für einen temporären Datensatz. Das System ändert die Disposition auf PASS
- JCL: Behandelt DCB-Parameter dynamisch
- JCL: Verbesserte SUM FIELDS-Ausgaben für falsche Werte
- JCL: CommonDDUtils: :GetContent sucht jetzt im Katalog nach RecordSize
- JCL: Liest die RDW/RecordSize-Attribute aus dem Katalog bei der Datensatzerstellung
- JCL: Unterstützung für DCB=.MYDD hinzugefügt, um DCB-Parameter eines DD im selben Jobschritt in ein anderes zu kopieren
- JCL: Verbessertes System zur Vererbung von Datensatzgrößen
- JCL: Exklusive Datensatzsperrung (Redis) hinzugefügt
- Redis: SSL-Unterstützung für den Standalone-Modus hinzugefügt
- Redis: Synchronisierte Redis-Sperrenanzahl mit Sperre hinzugefügt
- Redis: Unterstützte Pool-Parameter für Redis-Sperre
- Redis: Optimierte Metadaten-Aktualisierung mit Redis
- Redis: Verbesserte Redis-Cluster-Unterstützung
- Verbesserung bei offenen Schlössern im IO-Modus
- Verbesserte Leistung von Datensätzen, Sperren und Löschen ungenutzter Sperren
- Verbesserter Pfad des Datensatzes beim Aufheben der Registrierung der Datei
- Die Invalidierung des Pre-Fetch-Fenster-Cache wurde verbessert

- Unterstützung für die Verwendung von Thread-Safe-Utility-Datenquellenanbietern hinzugefügt
- Verbesserte DatasetState-Nulitätsprüfung
- Verbesserte Unterstützung dafür, dass bereits geöffnete Datensätze nicht erneut geöffnet werden
- Robustheit für den endgültigen Betrieb des Auftrags wurde hinzugefügt
- Verbesserte Unterstützung für Indizes, Reihenfolge für Schlüssel, wodurch Duplikate möglich sind
- Verbesserte Unterstützung für die Reihenfolge der Serialisierung von Skip-Listen
- Es wurde Unterstützung für die Debug-Dump-Funktion hinzugefügt, um Probleme mit der Reihenfolge der Indizes zu diagnostizieren
- Verbesserte Unterstützung für die Aktualisierung von Metadaten
- Verbesserte Unterstützung für Blusam Bulk Read

## ALS 400

### Neue Features

- Erstellt eine Registrierung im Anwendungskontext
- Support für das DSPF-Schlüsselwort CLRL (NO) Support die Überwachung von Datensatzsperrern
- Support für Keyed DataQueue
- Support für INQUIRY-Nachrichten für Batch-Jobs
- Unterstützung für die vom Programm beschriebene Druckerdatei für AS400 COBOL hinzugefügt
- Verarbeitet den Befehl RMVJOBSCDE cl
- Verbesserung für RUNSQL/DLYJOB
- CHKOBJ: Es wird ein älterer Fehlercode für den Parameter LIB ausgelöst
- SNDPGMMMSG: Unterstützt Zeichenkettenparameter
- RTVDTAARA: Verbesserte Teilzeichenfolge in LDA
- DSPFD: Der unterstützte FILE-Parameter wurde für einen bestimmten Dateinamen hinzugefügt
- RUNQRY: Support für SQL-Datei in QRY PARAM
- CRTDUPOB: Support für das Kopieren der Daten zwischen Datenbereichen
- SBMJOB: Konvertiert die Anweisung zur Verwendung JobQueueManager
- OPNQRYF: Unterstützung für die Qtemp-Bibliothek hinzugefügt
- CRTDUPOBJ: Verbesserte Logik für das Kopieren von Partitionsinhalten



- CRTDUPOBJ: Unterstützung für Qtemp für Views hinzugefügt
- RTVSYSVAL: Support für den SYSVAL-Wert, QDATFMT im CL-Befehl
- CHKOBJ: Unterstützung für OUTQ hinzugefügt
- RTVJOBA: Unterstützt den SWS-Parameter
- SNDPGMMMSG und RCVMSG: Zusätzliche unterstützte Parameter MSGF, MSGFLIB, MSGDTA, MSGTYPE, KEYVAR, MSGKEY, MSGID

## Verbesserungen

- Verbesserte Unterstützung für WORKSTATION-I/O-Karten
- Verbesserter Umgang mit festgelegter Nachricht, die vorherige Nachricht überlagert
- Unterstützt zusätzliche Nachrichteninformationen auf Array-Messageline
- Verbesserter eigenständiger Array-Wrapper-Zugriff innerhalb von EVAL, SortA und Bilddateien
- Verbessern Sie die DAOs-Reinigung, wenn die Online-Anwendung endet
- Unterstützung für zusätzliche Datumsformate hinzugefügt und die Handhabung von Zeichenketteneingaben verbessert
- Die CVTDAT-Behandlung von SYSVAL wurde verbessert, indem die Systemwert-Hilfsklasse Decode und Build-Parameter aus dem CL-Befehl hinzugefügt wurden SbmJob
- Das Paket com.netffective.bluage.gapwalk.rt.blu4iv wurde aus dem Komponentenscan entfernt gapwalk-cl-command
- Die Unterstützung vordefinierter Nachrichten für die Message Queue API wurde verbessert
- Die Unterstützung retrieveSubfileRecord für Datensätze, die in einem anderen Programm geschrieben wurden, wurde verbessert
- Die Unterstützung von Sofortnachrichten für die Message Queue API wurde verbessert
- Die Handhabung des lokalen Datenbereichs beim Einreichen eines Jobs wurde verbessert
- Startet JobQueues automatisch, wenn der Server gestartet wird
- Verwendet die ApplicationContext-Konfiguration, um Parameter für SBMJOB zu dekodieren
- Verbesserung der vom System bereitgestellten Fehlermeldungen
- Ermöglicht RTVMSG, in verschachtelten Unterverzeichnissen nach .properties-Dateien zu suchen
- Verwaltet das Zurücksetzen von Entitäten, die an fehlerhafte/ungültige Zeiger gebunden sind
- Es wurde verbessert MessageHandlingBuilder , dass MsgID und MsgFile Name als Zeichenketten für RCVMSG angezeigt werden

- Die withMsgFile Name-Methode der Message Queuing-API wurde verbessert
- Verbesserter Sperrmechanismus für den Datenbereich
- RTVMBRD: Support für Klein- und Großbuchstaben für den Parameter FILE
- CRTDUPOBJ: Verbesserte Handhabung von Ansichten
- CPYTOSTMF: Verbesserte Handhabung von Verbindungen
- CPYF: Verbesserung bei der Verarbeitung von Verzeichnisnamen beim Kopieren aus einer Flatfile
- RCVF: Behandelt DEV/RCDFMT-Parameter und die Transformation von RCDFMT für Groovy und Java ordnungsgemäß
- RCVF: Behandelt nachfolgende Aufrufe und verhindert, dass der Cursor zurückgesetzt wird
- CPYF: Unterstützung für das Schreiben aus flachen Dateien hinzugefügt
- CRTDUPOBJ: Die Behandlung von neuen Objekten mit der Qtemp-Bibliothek wurde hinzugefügt
- CHGDTAARA: Die maximale Länge des Datenbereichs wurde von 256 auf 2000 erhöht
- SAVOBJ: Stellen Sie sicher, dass die gespeicherten Datensätze in der Reihenfolge ihrer Einfügung gespeichert sind
- RTVDTAARA: Werte abgerufen (dürfen nicht gekürzt werden)
- CHKOBJ: Gibt korrekte Monitormeldungen zurück, wenn das Mitglied nicht existiert
- RTVDTAARA: Unterstützung für LDA-Substring hinzugefügt
- RTVDTAARA: Gibt Leerzeichen bis zur Länge der im RTNVAR-Parameter angegebenen Variablen zurück
- RTVDTAARA: Unterstützt Integer-Parameter für Start und Länge und unterstützt das neueste Transformationsformat
- CHGDTAARA: Unterstützung für Parameter hinzugefügt, die Unter- und Obergrenzen enthalten
- CHKOBJ: Behandelt den VIEW-Wert für den Parameter-Objektyp
- CHKOBJ: Das Ergebnis wird unabhängig vom Element auf „true“ gesetzt, wenn die Ansicht existiert

## Transversale Fähigkeiten

### Neue Features

- Übernimmt die Generierung von Berichten in TXT-Dateien
- Die Datenquelleneigenschaft CurrentSchema XA wurde dem Secret Manager hinzugefügt

- Fügen Sie die YAML-Eigenschaft `database.cursor.raise.already.opened.error` hinzu, damit das Framework den SQLCODE-Fehler 502 auslöst, wenn der bereits geöffnete Cursor geöffnet wird

## Verbesserungen

- Gapwalk-Poms zu AWS Blu Age auf Amazon EC2 EC2-Verpackungen hinzugefügt
- Verwendet standardmäßig das neue Signal-Handler-Paradigma
- Fügt Unterstützung für Sperren hinzu, wenn die Disposition MOD oder OLD ist
- Cache zum Speichern von Datums- und Uhrzeitmustern in der Datenbank hinzugefügt
- Verbesserte Prüffunktion von `PackedType`
- Verbessern Sie die `DataUtils` `.setTo`-Funktionen für Datensätze mit `VariableSizeArray`
- Behandelt die MQ SYNCPOINT-Option als Run-Unit
- Das Framework wurde aktiviert, um SQLCODE bei einer Rollback-Transaktion festzulegen
- Automatischer Treiberklassenname gemäß dem geheimen Engine-Schlüssel hinzugefügt
- Timeout für Programm/Transaktion
- Stellen Sie die Cursorposition nach dem Rollback beim Zugriff auf den Cursor wieder her

## Drittanbieter

- SnakeYAML, Redisson und Amazon SDK aktualisieren, entfernen `YamlBeans` (CVE-2023-25857, CVE-2023-24621, CVE-2023-42809, CVE-2023-44487 mindern)

## Version 3.9.0 der Modernisierungstools

### zOS

#### Verbesserungen

- Verbesserte Unterstützung für XML-TEXT als Quelle für ein Ziel vom Typ `String`
- Verbesserter STM-zu-UML-Workflow zur Unterstützung von X/ (Y/Z) -Divisionsmustern
- JHDB DB: Akzeptiert den ROLLBACK-Aufruf vor jedem Datenbankupdate
- JHDB DB: Akzeptiert ROLLBACK, auch wenn die Transaktion beendet ist (NOP)
- JCL: Verbesserte Funktion zur schrittweisen Validierung

- SORT: Behandelt die SUM-Funktion mit negativen Zonendecimalwerten
- COBOL: Fügt Unterstützung für das Escapen von einfachen/doppelten Anführungszeichen in Zeichenkettenliteralen hinzu

## AS400

### Verbesserungen

- Die Verarbeitung des Bearbeitungscode X mit der integrierten Funktion %editc wurde verbessert, indem führende Nullen hinzugefügt wurden
- Verbesserte Behandlung des Anfangswerts von Eingabefeldern
- Aktionstasten für Hilfsdialoge hinzugefügt
- Fußzeileneintrag der dynamischen Tabelle, der unten angezeigt wird
- Hat den START-Befehl ohne KEY PHASE für Dateien verarbeitet, die einen tatsächlichen RECORD-KEY angeben
- Es wurde ein Standardwert für die Typen Float und NumberUtils: :pow hinzugefügt
- Unterstützung für die Definition einer Variablen mit LIKE (IN) hinzugefügt
- Die FOR-Schleifenbehandlung wurde aktualisiert, um das Auslassen optionaler Elemente zu unterstützen
- Das RPG-Parsing wurde aktualisiert, um Datensätze mit dem CTDATA-Array-Namen zu verknüpfen
- Verbesserter Umgang mit Indikatoren für CabXX-Anweisungen
- Unterstützt optionale Parameter für das COMMIT-Schlüsselwort
- Verbesserte Unterstützung für FORMAT-Schlüsselwörter in LF
- Verwalteter LOOKUP-Operationscode mit hohen und gleichen (oder niedrigen und gleichen) Indikatoren
- Behandelte PF-Schlüsselname, der in doppelten Anführungszeichen deklariert ist
- Die Behandlung von EDTCDE X wurde verbessert, sodass führende Nullen nicht unterdrückt werden
- Verbesserte Unterstützung für MSGCON in einer Druckerdatei, die keine unbenannten Labels generiert
- Der Feldinhalt wird von mehreren Datenstrukturen gemeinsam genutzt

- Behandelte den ERRSFL-Parameter in Kombination mit SFLMSG/SFLMSGID
- Der Umfang des vollständigen kostenlosen Rollenspiels vor der Proc-Deklaration wurde verbessert
- Die Spezifikation für das Parsen von Conditioned Control wurde hinzugefügt
- Verbesserte Unterstützung für die Methode setErrSfl () in Dataholdermapper
- Verbesserte Typauflösung für intern erstellte Variablen
- Verbesserte Unterstützung für den Z-ADD-Opcode
- Die Behandlung von konstanten Feldern mit DFT-Werten wurde verbessert
- Verbessert die Unterstützung von Integer-Feldern im Programmstatus ds
- Hat die Indikatorzuweisung in den ENTRY-Parametern abgewickelt
- Der Filter von Schlüsselwörtern, die über das Schlüsselwort ref/reffield weitergegeben werden, wurde verbessert
- Unbenannte Datenstruktur wird unterstützt DataArea
- Verbesserte Handhabung des Zeiger-Datentyps
- Behandelte Array-Elemente, die zur Definition von Variablen mit dem Schlüsselwort LIKE verwendet werden, unterstützen den Array-Zugriff im Ausgabefeld
- Verbesserte Unterstützung für numerische Zahlen mit Vorzeichen, wobei nur Ziffern angezeigt werden
- Unterstützte logische Beziehung auf O-Karte
- Testfall für %CHAR auf alphanumerisch
- Das Schlüsselwort main wird für die Steuerspezifikation unterstützt
- EDTCDE mit zwei Parametern in der Druckerdatei
- Verbessertes RPG-Parsing FullFree
- Die dynamische Tabelle wurde verbessert, um sicherzustellen, dass die Fußzeile korrekt positioniert ist
- Unterstützung für die Initialisierung numerischer Typen mit ALL-Bildkonstanten hinzugefügt
- Die Handhabung mehrerer logischer RPG-Dateien, die auf dieselbe physische Datei verweisen, wurde verbessert
- Verbessern Sie die Erkennung modifizierter Felder in einem modernen Bildschirm
- Modale Synchronisation mit dynamischen Feldern
- Die Behandlung von numerischen Feldern, die nur mit Vorzeichen ausgegeben werden, wurde verbessert

- Verbesserung der Unterstützung von WORKSTATION-I/O-Karten

## Transversale Funktionen

### Neue Features

- Data Migrator-Tool: Die `ebcdicFilesWith VarcharIn` VB-Eigenschaft wurde hinzugefügt, um die Berücksichtigung der 2-Byte-Länge von `VARCHAR` beim Lesen von Bytes zu ermöglichen
- Es wurde eine allgemeine API zur Protokollierung von Fehlern implementiert
- Implementierung `BluAgeErrorDictionaryUtils` und Verwendung einer gemeinsamen API zur Protokollierung von Fehlern und/oder Informationen in `COBOL2Model`, `CycleBuilder RPG`, `Definitions2Model` und `FieldsProcessor`
- Verbesserte SQL-Grammatik zur Unterstützung verschiedener Definitionen von Isolationsklauseln

### Verbesserungen

- Angular-Version auf v16 aktualisiert
- Angular: Die `ajv`-Version wurde von 6 auf 8.9 aktualisiert

### Drittanbieter

- Groovy wurde auf Version 2.4.15 aktualisiert

## Versionshinweise 3.8.0

Diese Version von AWS Blu Age Runtime and Modernization Tools konzentriert sich auf mehrere bereichsübergreifende Verbesserungen des Produkts zur Verbesserung der Qualität und Sicherheit sowie auf Leistungsverbesserungen beim Zwischenspeichern und die Vereinheitlichung der Befehlsunterstützung in einer einzigen Distribution. Einige der wichtigsten Funktionen und Änderungen in dieser Version sind:

- Versionsupgrade von Spring 2.5 auf Spring 2.7, wodurch der Wartungssupport, die Leistung und die Sicherheit der Plattform erhöht werden.
- Vereinheitlichung der Unterstützung von mehr als 82 CL-Befehlen als Teil der over-the-counter Distribution, um die Nutzung und Bereitstellung modernisierter Anwendungen zu erleichtern, die zuvor CL-Scripting nutzten.

- Neue APIs für einen besseren Betrieb und eine bessere Interaktion mit BluSAM-Datensätzen, wie z. B. der integrierte Import in den Managed Service und die Möglichkeit, Datensatz-Metadateninformationen aufzulisten, stehen zur Verfügung.
- Leistungsverbesserungen und Ausweitung der Nutzung von Redis, einschließlich Verfügbarkeit im Clustermodus, Hochverfügbarkeitsdatenabruf und Standardisierung der Verwendung von Geheimnissen.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 3.8.0

### zOS

#### Neue Features

- Behandlung der Schlüsseldefinition als Zeichenfolge für DynamicFileBuilder
- DFSORT: Unterstützung für mehrere Elemente in der Grammatikinitialisierung von OUTFIL TRAILER1 + DFSORT hinzugefügt
- CommonDDUtils-Tool: Umgang mit der Datensatzgröße in In-Stream-Daten
- Indizierte Datei: Behandlung der GENKEY-Option

#### Verbesserungen

- Externalisierte BluSAM-Ladedienste in einem separaten Jar
- Unterstützung für die Einrichtung eines Speicherorts für temporäre Dateien hinzugefügt
- Verbesserte gemeinsame Cache-Mechanismen für Fälle mit mehreren Knoten
- Nutzung gemeinsam genutzter Caches: IDCAMS verifizieren die Optimierung
- Verbessern Sie die ROWID-Injektion für Embedded Select
- JCL: Jede In-Stream-Jobprozedur wird jetzt in einer eigenen Groovy-Datei generiert
- Stellen Sie sicher, dass auf IDCAMS card-demo-v JCL-Karten 2 abgedeckt sind
- BluSAM: Vermeiden Sie doppeltes Aufwärmen, wenn Sie mehrere Instanzen verwenden
- Reduzierter Speicherbedarf bei der Cache-Flüssigkeitszufuhr
- Unterstützung für Jedis-Pool-Konfigurationen

- Zeilentrenner zum Stream hinzugefügt, falls er bei der Dateiverkettung verwendet wird
- Support für EBCDIC-Karten + Blockkommentare (/.../) im IDCAMS-Hilfsprogramm
- Datenbankunterstützungsabfrage: Unterstützung für Doppelbyte-Strings bei der Konvertierung von Level49 nach SQL
- DFSORT-Grammatik: implementiert 17 Steueranweisungen und Integration von 2 davon (OMIT/INCLUDE)
- Verbessert das Abrufen von INFUTILB durch GRAPHIC-Spalten
- Support für das Lesen von Dateien mit variabler Größentabelle
- Support für ZonedType with Nibble signed, wobei das erste Bit des letzten Bytes 'E' ist
- DFSORT/ICETOOL fügt Unterstützung für das Argument NOMATCH =(..) hinzu, falls ein Datensatz keiner der CHANGE-Suchkonstanten entspricht
- Redis-Cluster-Kompatibilität
- Verarbeitung des Jobstatus (Fehlgeschlagen) basierend auf dem Groovy-Exit-Code
- Verbesserte CICS SYNCPOINT ROLLBACK-Unterstützung.
- Pre-Fetch-Fenster zur Optimierung der Redis-Cache-Nutzung
- JCL/GROOVY: Erbt die IsRDW-Eigenschaft aus dem Datensatz des vorherigen Schritts, wenn DISP =(, PASS)
- Behandlung von Teilkopien von Daten mit einem Array variabler Größe

## AS400

### Neue Features

- Support für I/O-Karten für Displaydateien
- Support für zusätzliche Nachrichteninformationen für die DSPF-Schlüsselwörter ERRMSGID und CHKMSGID
- Support für mehrere Fehlermeldungen auf dem Frontend-Bildschirm
- Die Unterstützung von 82 CL-Befehlen innerhalb der Anwendung wurde hinzugefügt oder verbessert gapwalk-cl-command

### Verbesserungen

- Verbesserte Unterstützung für DELETE und READ unter Commitment-Kontrolle



- ConvertDate innerhalb des eingebauten %dec
- Erzwungene XSS-Sicherheitsheader
- Verbesserte Robustheit und Konsistenz der STM-Generierung (bessere Handhabung von: Fortsetzungszeilen in freier Form im RPG, Kommas für Dezimalteile, Freiformblöcke in Definition/Deklaration)
- DataHolderMapper Verbesserte Generierung
- Zusätzliche Robustheit und Änderungsumfang in DataAreaFactory
- Die Fokusverlagerung bei der Tabulatortaste wurde verbessert
- Die Leistung bei der Generierung von Jasper-Berichten wurde verbessert
- Verbesserte Dezimalanzeige mit Auffüllung von Nullen
- Verbesserte Unterstützung für das ROW/COL-Feld in INFDS
- Verbessert die Unterstützung für geänderte Felder auf dem Bildschirm
- Es wurden Getter für den Namen und den Pfad des generierten Berichts hinzugefügt
- Die Länge der Datenwarteschlange wurde verbessert
- Verbesserte Autokonfiguration von Job Queues, um den neuen Standards in Spring Boot 2.7 zu entsprechen
- Verbesserte Workstation-Updates für mehrere gleichzeitige Sitzungen

## Transversale Funktionen

### Neue Features

- Support für No Invalid Data Tolerance für Packed
- Paginierung/Filterung zur Liste der Datensatz-Endpunkte hinzugefügt

### Verbesserungen

- Verbesserte Strategie zur Transformation von ORACLE-Abfragen beim Vergleich von Spalten mit leeren Zeichenketten
- Umgang mit BLOB DB2 mit den Hilfsprogrammen DSNTDP und INFUTILB. BLOB DB2 wurden jetzt auf Postgres vom Typ BYTEA modernisiert.
- Verbesserung beim Löschen des letzten Cursorelements
- Verbesserte Unterstützung für das Löschen von RRDS-Dateien

- Die Leistung von AWS Blusam Secret wurde verbessert
- Verbessertes Umgang mit Datenbankverbindungen im SQL-Framework
- Standardisierte geheime AWS Manager-Schlüssel für mehrere Datenquellen
- Behebung von Leistungsregressionen
- Verbesserte Checkfunktion für PackedType
- Verbessertes Umgang mit LOW-VALUE für PackedType
- Verbesserte Spring Security-Verpackung für Cognito Connection
- Die CodeshiftPoint-Kodierung und -Dekodierung wird nicht auf DB2-Zieldatenbanken angewendet

#### Drittanbieter

- Spring Boot-Upgrade von 2.5 auf 2.7

## Version 3.8.0 der Modernisierungstools

### zOS

#### Neue Features

- JCL: Verarbeitung von Streams mit Zeilenrückgabe „\ r“

#### Verbesserungen

- Verbesserte Protokollierung, um eine Division durch Null bei der Modernisierung einer DIVIDE mit ON SIZE ERROR-Klausel zu verhindern
- JCL: Verbesserte Unterstützung für das Aufrufen einer Prozedur in einer Prozedur
- Support für das Schlüsselwort OF im CICS-Befehl FORMATIME bei mehrdeutigen Feldern
- JCL: Unterstützung für das Zeichen Å¥ in Variablen
- JCL: Berechnung von RC auf der Grundlage früherer Schritte
- Vergleich von Bytes anstelle von Zeichenketten, wenn PL1 SUBSTR verwendet wird
- Verbesserung der Initialisierung von multidimensionalen Arrays aus einer einzigen Quelle
- Verbessertes Parsen von COBOL, wenn es sich um eine einzelne SQL-Abfrage in einem IF-Block handelt

## AS400

### Neue Features

- Support für verschachtelte IF-Anweisungen in CL
- Verbesserte Unterstützung für ENDDO-Anweisungen in RPG Freeform

### Verbesserungen

- Verbesserte Unterstützung für Conditioning Control Level
- Verbesserte Rückgabe des Prototyps mit LIKE
- Verbesserte Unterstützung für die Handhabung von Funktionen %Monate, %Jahr, %Tage
- Support der Hilfsfunktion für den gesamten Bildschirm
- Behandlung figurativer LEERZEICHEN, die als Parameter übergeben wurden
- Verbesserung des Ausdrucks EVAL mit dem Operator „“
- Behandlung des START-Befehls ohne KEY PHASE
- Verbesserung beim Umgang mit dem Schlüsselwort LIKEREK
- Verbesserung bei unbenannten Unterfeldern
- Verbesserung des Verfahrens zur Rückgabe eines unsignierten Typs
- Verbesserte Unterstützung für den RESET-Vorgang (kostenloses RPG), die integrierten Funktionen %CHAR und %DEC
- Verbesserung der eingebauten Funktion %LOOKUPXX
- Verbesserte Unterstützung für das Schlüsselwort LIKEDS bei Verfahren ohne Prototyp
- Umgang mit dem Schlüsselwort-Array-Typ Dim (VAR, AUTO)
- Verbesserte Unterstützung für XFOOT
- COBOL: verbesserte Unterstützung für RAMES-Felder
- CL: unterstützt die Bedingung while (true)
- Die Behandlung von eigenständigen Arrays mit dem Schlüsselwort LIKE wurde verbessert
- Verbesserung der integrierten Funktion %INT
- Verbessertes RPG Full Free Parsing
- Verbesserte Unterstützung für Arrays in der Verknüpfung
- CL2GROOVY: Select-Erklärung Support

- Verbesserung des DSPF-Schlüsselworts „ERRMSGID“
- Die Behandlung der Initialisierung von Bytes mit führenden Nullen wurde verbessert
- Verbesserung von AuthorizedValues für numerische Felder
- Behandlung von Extender H für die EVAL-Anweisung in freier Form
- CL to Groovy: Support Teilzeichenfolge von LDA
- Die Unterstützung für RESET auf einem Datensatz wurde verbessert
- Die Handhabung von EDTCDE und EDTWRD mit Referenzen wurde verbessert
- Verbesserte Zuordnung von Eingabefeldern mit DDS-Feldern
- Verbesserte Unterstützung für MOVEA-Zeichen in ein IN-Array
- Verbesserung des Prototyps mit dem Schlüsselwort LIKEDS
- Verbesserte Unterstützung für das DSPF-Schlüsselwort DSPATR
- Verbessertes Parsen von D-Karten mit +/-
- Robustheit bei Programmaufrufen hinzugefügt
- Der Prozess zur Problemlösung wurde robuster

## Transversale Fähigkeiten

### Verbesserungen

- FrontEnd: Simuliert das Einfügeereignis für die IME-Eingabe

### Drittanbieter

- Spring Boot-Upgrade von 2.5 auf 2.7

## Versionshinweise 3.7.0

Diese Version von AWS Blu Age Runtime and Modernization Tools umfasst hauptsächlich Verbesserungen zur besseren Unterstützung von Befehlen und Dienstprogrammen, Funktionen zur Integration mit AWS Secrets Manager und neue Überwachungsfunktionen. Einige der wichtigsten Änderungen in dieser Version sind:

- Mehrere Laufzeitkomponenten können jetzt AWS Secrets Manager verwenden, um die Sicherheitseinstellungen modernisierter Anwendungen zu verbessern, hauptsächlich im

Zusammenhang mit Dienstprogrammen, Datenquellen, Redis für TS-Warteschlangen, BluSam Cache und Sperren.

- Überwachungsendpunkt, der das Abrufen von Transaktions-, Batch- und JVM-Metriken zur Optimierung der Ressourcennutzung und zur Betriebsverwaltung ermöglicht, z. B. Status, Dauer, Volumen und andere.
- Neue Funktionen zur Unterstützung von IBM MQ-Aufrufen in RPG und erweiterte Abdeckung für JCL SORT- und IDCAMS-Transformationen.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 3.7.0

### Themen

- [zOS](#)
- [ALS 400](#)
- [Transversale Fähigkeiten](#)

### zOS

#### Neue Features

- Verbessern Sie das Parsen von Abfragen im Zusammenhang mit Programm-Hilfsprogrammen, indem Sie SQL-ähnliche Grammatik verwenden. (V7-9401)
- Behandelt ein indiziertes Array mit variabler Größe beim Offset (V7-9904)
- Support die INSERT SQL TIME-Spalte in DB2 im 24:00:00-Stundenformat (V7-10023)
- Support INSERT-SQL-Abfragen aus Arrays mit den Optionen FOR ROWS und ATOMIC (V7-10105)
- JCL SORT — Erweiterung zur Unterstützung von OUTREC TranscodeTool mit IFTHEN (V7-10124)
- JCL SORT - fügt Unterstützung für das DATE-Schlüsselwort im OUTREC-Befehl hinzu (V7-10125)
- JCL — fügt Unterstützung für In-Stream-Prozeduren hinzu (V7-10223)

#### Verbesserungen

- Ein Datensatz, der mit der Disposition „PASS“ gekennzeichnet ist, sollte für alle Arbeitsschritte verfügbar sein (V7-9504)
- Support das JCL-Attribut SCHENV (V7-9570)
- Support von SEND mit CTLCHAR-Option (V7-9714)
- COBOL - Behandelt verschiedene Zeilentrennzeichen-Zeichensätze in ACCEPT-Anweisungen (V7-9875)
- Vermeiden Sie mehrfaches Rollback (V7-9958)
- Erlaubt die Verwendung der MOD-Disposition zum Anhängen am Ende von GDG-Dateien (V7-10031)
- Optimierung: PutAll-Refactoring (V7-10063)
- PutAll Refactoring: Paginierung hinzufügen (V7-10063)
- Machen Sie das Lese-Timeout für den Jedis-Client konfigurierbar (V7-10063)
- UseSsl Unterstützung für den Standalone-Modus (V7-10114)
- Support von EIBDS nach erfolgreichem Öffnen der Datei (V7-10147)
- Support von EIBDS nach einer Dateikontrollanfrage (V7-10147)
- Verbessern Sie die CICS SYNCPOINT-Unterstützung (V7-10187)
- BluesamRedisSerializer: Problem mit MetadataPersistence (V7-10202)
- Support von Redis AWS Secrets Manager für TS-Warteschlangen (V7-10204)
- Support von JCLBCICS beim Anpassen der DD-Namensgröße (V7-10224)
- Fügt Unterstützung für absolute Pfade in der IDCAMS DELETE-Anweisung hinzu (V7-10308)

## ALS 400

### Neue Features

- Implementierung der Hilfsfunktion für AS400-Bildschirme (V7-9673)

### Verbesserungen

- Anzahl der Datensätze in INFDS (V7-9377)

## Transversale Fähigkeiten

### Neue Features

- Support für Runtime auf EC2 zum Senden von Protokollen an Amazon CloudWatch (D87990246)
- Neuer Endpunkt zum Abrufen von Metriken zu Batches, Transaktionen und JVM hinzugefügt (D88393832)

## Verbesserungen

- Unterstützte Datenquellen AWS Secrets Manager für das Hilfsprogramm pgm (V7-9570)
- Db2-Unterstützung für DSNUTILB DISCARD (V7-9798) hinzugefügt
- Support für das Schreiben in den Logger statt in den Standardsystemausgabestream in den Standard-SYSPRINT- und SYSPUNCH-Dateien (V7-10098)
- Support BluSam Redis-Cache und sperrt Verbindungseigenschaften in AWS Secrets Manager (V7-10238)
- Support für SSL-Verbindungen auf Db2 XA AWS Secret (V7-10258)
- Die Metadaten für IDCAMS REPRO und VERIFY (V7-10281) wurden aktualisiert
- Verbessertes IDCAMS-Rückgabecode-Management am Abend (V7-10307)

## Version 3.7.0 der Modernisierungstools

### Themen

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

### zOS

#### Neue Features

- PLI — Verbesserte Zuweisung für Array-Querschnitte und zweidimensionale Arrays (V7-9830)

### AS400

#### Neue Features

- Umgang mit Kontrollstandsanzeigen (V7-9227)
- Support für den EXTNAME-Parameter \*INPUT (V7-9897)

- Verbessertes Goto Rewriting: Support für Tags in SELECT OTHER-Anweisungen (V7-9973)
- Support das REFSHIT DSPF-Schlüsselwort (V7-10049)

## Verbesserungen

- Verbesserung beim Umgang mit dem Dateibeschreibungsschlüsselwort EXTIND (\*Linux) (V7-7404)
- Verbesserte Transformation von SQLDDS-Dateien (V7-7687)
- Dateiobjekte werden nicht mehr für AS400-Dateien generiert (V7-9062)
- Verbesserte Behandlung des Dateibeschreibungsschlüsselworts EXTDESC (V7-9268)
- Verbesserte Handhabung von %CHAR builtin (V7-9311)
- Verbesserte Unterstützung für Pagedown beim letzten Datensatz ohne SFLEND (V7-9322)
- Verbesserte Unterstützung für Datenstrukturen mit Präfix (V7-9436)
- Support für mit %SIZE definierte Dimension (V7-9472)
- Support für die Behandlung von PF-Feldnamen, die in doppelten Anführungszeichen deklariert sind (V7-9557)
- Verbesserte Dateioperation — Groß- und Kleinschreibung wird nicht berücksichtigt (V7-9785)
- Support für Felder, die mit \*USER initialisiert sind (V7-9806)
- Support für den COMP-Typ in AS400 (V7-9840)
- Verbessertes COBOL400-Parsing bei (Not) (V7-9922) InvalidKey
- Verbesserte Handhabung des SCAN-Vorgangs (V7-9971)
- Verbesserte Unterstützung des GOTO-Opcodes (V7-9973)
- Verbesserte Handhabung der EXCEPT-Operation (V7-9977)
- Verbesserte Präfixunterstützung (V7-10000)
- Support für MQ-Aufrufe in RPG (V7-10007)
- Die integrierte %LOOKUP-Funktion (verschlüsselte Array-Datenstruktur) wurde verbessert (V7-10022)
- Support für den Vorgang „Alle schließen“ (V7-10036)
- Support für die SQLDDS-Anweisung UPDATE AS ROW CHANGE (V7-10051)
- Verbesserung beim Umgang mit dem Literalwerttyp Long (V7-10073)
- Verbesserte RPG-Grammatik (Verwendung des Schlüsselworts INZ als Name der Subroutine) (V7-10074)



- Verbesserte RPG-Grammatik zur Unterstützung numerischer Werte mit leerem Bruchteil (V7-10077)
- Verbesserte Unterstützung für Felder, die von CL und externer Datei gemeinsam genutzt werden (V7-10081)
- Verbesserte Unterstützung für DDS-Bedingungsindikatoren (V7-10084)
- Support für den DDS-Binärtyp mit COBOL-Programmen (V7-10100)
- Verbesserte Namenskollision mit Verknüpfung (V7-10109)
- Support für das Mischen von Haupt- und Exportverfahren (V7-10112)
- Verbesserte Unterstützung für DataStructure in einem Unterverfahren (V7-10113)
- Verbesserte Unterstützung von CLEAR (V7-10126)
- Verbesserte Unterstützung von DO Loop (V7-10134)
- Support von SQLTYPE im vollständig kostenlosen RPG (V7-10151)
- Verbessertes Parsen der Bedingungen für das DDS-Schlüsselwort (V7-10155)
- Verbesserte DSL-Generierung (V7-10163)
- Verbesserung für ProcessIndicators, wenn es sich bei der Bedingung um einen binären Ausdruck handelt. (V7-10164)
- Der Zustand GoTos mit Else wurde verbessert (V7-10168)
- Support für den Typ Time und Timestamp in DSPF (V7-10173)
- Verbessertes Parsen der Fortsetzungszeile für DDS (V7-10183)
- COBOL-Unterstützung für RENAMES FLD OF RECORD (V7-10195)
- Verbessertes Parsen von bedingten Indikatoren in DSPF-Feldern (V7-10221)
- Support das Parsen des DDS-Schlüsselworts NOALTSEQ (V7-10288)
- Support-Hilfemenü und versteckte Felder (V7-10314)
- Verbesserte Überprüfung der Richtigkeit von Schlüsselwörtern in der DSPF-Hilfe (V7-10328)
- Es werden nicht mehr alle Schlüsselwörter im Referenzfeld weitergegeben (V7-10347)

## Transversale Fähigkeiten

### Neue Features

- Data Migrator — Umgang mit CLOB-Daten (V7-9665)

## Verbesserungen

- Übertragung der JCL-Eigenschaft SCHENV von JOB zur PROC GROOVY-Definition durch (V7-10225) JobContext
- FrontEnd - Anpassung der Fenstergröße bei fehlendem Rand (V7-10358)

## Versionshinweise 3.6.0

Diese Version von AWS Blu Age Runtime and Modernization Tools bietet neue Funktionen für ältere ZoS- und AS400-Migrationen, die hauptsächlich auf die Erweiterung der CICS-Unterstützungsmechanismen, die Ergänzung der JCL-Funktionen, die Optimierung der Leistung bei gleichzeitigen und umfangreichen Funktionen und das Hinzufügen von Funktionen ausgerichtet sind. multi-data-source Einige der wichtigsten Änderungen in dieser Version sind:

- Verbesserung der dynamischen Dateiverwaltung in JCL, Erweiterung der aktuellen Anweisungen und Verwaltung verketteter Datensätze, Ausführung mehrerer Anweisungen in einem einzigen Block und Datenübertragung von Batches zu Programmen.
- Verbesserte Unterstützung mehrerer CICS-Befehle, einschließlich der Abfrage mehrerer CICS-Ressourcentypen.
- Die Möglichkeit, bei der Verwendung von Blu Age Runtime Utilities unterschiedliche Datenbanken zu verwenden, eignet sich am besten für Szenarien, in denen Geschäftsdaten auf mehrere Quellen verteilt sind.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 3.6.0

### Themen

- [zOS](#)
- [ALS 400](#)
- [Transversale Fähigkeiten](#)

## zOS

### Neue Features

- JCL - DynamicFileBuilder - Verbesserte Verwaltung von Datei-Handles (V7-9408)
- Verbesserte Formatkonvertierung einiger integrierter SQL-DB2-Funktionen beim Aufruf des Dienstprogramms INFUTILB UNLOAD (V7-9554)
- Verbesserte multidimensionale PLI Array-Zuweisungen (V7-9592)
- Handhabung der Sysout-Umleitung zur Datei (V7-9992)

## Verbesserungen

- Hinzufügen der Triggerung von gespeicherten Prozeduren für DB2 RDBMS (V7-9155)
- SORT kümmert sich um die Konvertierung in das PDF-Format (V7-9286)
- JCL/GROOVY — Erweiterung der REPRO-Anweisung zur Unterstützung von DUMMY-Datensätzen (V7-9424)
- Verbessern Sie die CICS UNLOCK-Unterstützung (V7-9606)
- Behandelt die Standardwertgröße für Union (V7-9648)
- JCL/GROOVY handhaben unterschiedliche Terminierungen/Dispositionen in verketteten Datensätzen (V7-9653)
- pageSize für Blusam-Datensätze konfigurierbar machen (V7-9680)
- DSNUTIL — erlaubt das Laden von 24:00:00 als gültige ZEIT in DB2LUW (V7-9697)
- Support den Vergleich mit HOHEN WERTEN (0xff) in NumberUtils .ne ()/ NumberUtils.eq () (V7-9731)
- JCL/GROOVY - unterstützt DO... THEN-Schlüsselwörter in IDCAMS IF-THEN-ELSE-Klauseln zur Ausführung mehrerer Anweisungen in einem einzigen Block (V7-9750)
- Ungültiges JHDB hat ein Programm außerhalb von JHDB aufgerufen (V7-9782) BatchRunner
- Support Leerzeichen auf der SORT UTFIL-Steuerkarte (V7-9808)
- Verbessern Sie die CICS READ PREV-Unterstützung (V7-9845)
- Verbessern Sie den gleichzeitigen Zugriff auf Datensatzindizes (V7-9864)
- Verbessern Sie die CICS REWRITE-Unterstützung (V7-9873)
- COBOL — Unterstützung für mehrzeiliges SYSIN in ACCEPT-Anweisungen zur Übergabe von Daten aus einem Batch (JCL) an ein Programm (COBOL) (V7-9875)
- Groovy — Bessere Handhabung von AT-Dateierstellungsschritten (V7-9876)  
ConcatenatedFileConfiguration

- IDCAMS UTILITY - Behandlung der Anweisung DEFINE PATH (V7-9878)
- SORT BUILD — Passen Sie die TRAN-Option an und behandeln Sie implizite Leerzeichen (V7-9925)
- Verbessern Sie CICS DELETE mit der Unterstützung der GENERIC-Option (V7-9939)
- Verbessern Sie die Unterstützung für CICS STARTBR und ENDBR (V7-9952)
- Verbessern Sie die Leistung bei gleichzeitigem Zugriff (V7-9953)
- Verbessern Sie die Behandlung des Dateistatus beim Start (V7-9991)
- Groovy - Erlaubt den Aufruf von getDisposition () /get () /get () auf NormalTermination (V7-10012) AbnormalTermination ConcatenatedFileConfiguration

## ALS 400

### Neue Features

- Support externer Indikatoren für COMMIT-Schlüsselwörter (V7-6035)
- Setzen Sie die ReadC-Schleife nach dem SFLCTL-Schreiben zurück (V7-8061)
- Support der LR-Anzeige in CALL (V7-9250)
- Fügen Sie einen neuen Typ von dynamischem Feld (aufgeteilt) hinzu, um Eingabefelder in mehreren Zeilen zu verarbeiten (V7-9370)
- Support primäre/sekundäre Dateien (V7-9390)
- Der lokale Datenbereich wird jetzt beim Senden eines Jobs an den aufgerufenen Job übergeben (V7-9775)
- Support von QTEMP für den Datenbereich und Unterstützung der Wertschöpfung im Datenbereich. (V7-9916)
- Commitment Control — Unterstützung für das Aktivieren/Deaktivieren der Commitment Control (V7-9956)
- Support externer Indikatoren für COMMIT-Schlüsselwörter

### Verbesserungen

- Verbessern Sie die Anzeige von 0-Werten und EDTWRD (V7-8933)
- Support des DSPF-Schlüsselworts „CHKMSGID“ (V7-9125)
- SQL-Commit-Transaktion bei Batch-Beendigung (V7-9232)

- Verbessern Sie die Unterstützung der Schlüsselwörter EXPORT und IMPORT für Feld- und Datenstruktur (V7-9265)
- Support Kleinbuchstaben in DateHelper (V7-9461)
- Support die Konvertierung \*CYMD zu\*ISO (numerisch) (V7-9488)
- Verbessern Sie das Handle des integrierten %len für ein variierendes Feld (linke und rechte Seite eines Ausdrucks) (V7-9733)
- Verbessern Sie die Unterstützung für die integrierten Funktionen '%LOOKUPXX' XX („LE“, "LT“, "GE“, "GT") (V7-10064)

## Transversale Fähigkeiten

### Neue Features

- CICS — Verbesserung der Abfragetransaktion für den Optionsstatus (V7-9712)
- JCL — Verbessern Sie den Ladevorgang für Sysprint mit einer Systemausgangsdatei (V7-9797)
- CICS — Verbessern Sie INQUIRE TSQUEUE (V7-9823)
- CICS — Verbessern Sie das Inquire-Terminal für die Option Benutzer-ID (V7-9906)

### Verbesserungen

- Verbessern Sie die Handhabung des Vergleichs mit Blank (V7-8047)
- Verbessern Sie die Protokollierung für Jics und Blusam (V7-8847)
- Support erweiterte BMS-Attribute SOSI und das programmierte Symbol F8 für dynamische Felder (V7-8857)
- Behandeln Sie den Pufferüberlauf im Programmparameter (V7-9138)
- Verbessern Sie die Parallelität der Threads beim Schreiben von Blusam Locks Registry (V7-9505)
- Support die Konfiguration mehrerer Datenquellen für Utility-PGM (V7-9570)
- Blusam-Modus nur zum Sperren auf Datensatzebene (V7-9626)
- Stellen Sie sicher, dass die Persistenz der Metadaten einem Serverneustart standhält (V7-9748)
- Verbessern Sie die DAO-Bereinigung bei Ausnahmefällen (Browser schließen) (V7-9790)
- Support DummyFile für INFUTILB SYSPUNCH (V7-9799)
- Verbessern Sie die Unterstützung für negative Werte auf (V7-9935) NumericEditedType

## Version 3.6.0 der Modernisierungstools

### Themen

- [zOS](#)
- [ALS 400](#)
- [Transversale Fähigkeiten](#)

### zOS

#### Neue Features

- JCL - Verbesserte Protokollierung für das Ende des Vorgangs (V7-8509)
- PL1 — Verbesserte Generierung von Beuteln für den Datentyp (V7-8917) PakedLong
- JCL - Verbessert die Protokollierung für das Ende des Vorgangs, wenn die Datei die Markierung „Ende“ enthält/(V7-9509)
- PL1 - Verbesserte Unterstützung für GET EDIT mit Fixed-Point- und SYSIN-Stream (V7-9593)
- DB2 — Verbessert die Unterstützung für den Typ VARGRAPHIC DB2 (V7-9809)
- CICS — Verbessern Sie den Befehl QUERY SECURITY für die Option LOGMESSAGE (V7-9969)
- PL1 — Verbesserung der Beutelgenerierung für Charg/CharGraphic built-in (V7-9989)

#### Verbesserungen

- PL1- Verbesserte Unterstützung für das INCLUDEX-Schlüsselwort (V7-9588)
- PL/I — Behandelt das Schlüsselwort CHARGGRAPHIC als gültigen Parameter eines beliebigen Methodenaufrufs (V7-9589)
- Verbesserung der Auflösung von PL1-Hostvariablen, wenn sie mit bestimmten Zeichen @ # \$ § benannt werden. (V7-9654)
- COBOL - Support der Schlüsselwörter C01... C12 und S01... S05 als Parameter der WRITE ADVANCING-Anweisung beim Analyseschritt (V7-9669)

### ALS 400

#### Neue Features

- Support der SQL-DDS-Transformation in Analyzer (V7-7687)

- Automatisieren Sie die Erkennung von SQL-DDS-Dateien (V7-7687)
- Implementierung der SQL-DDS-Vorverarbeitung (V7-7687)
- Support das ALIGN-Schlüsselwort (V7-9254)
- Support für ExtName DSPF und Multi-Dim-Array (V7-9663)
- InvalidKey Unterstützungserklärungen zu COBOL WRITE (V7-9793)

## Verbesserungen

- Verbesserung des TESTB-Opcodes (V7-8865)
- Verbessern Sie die Unterstützung von DECFMT im Fokus (V7-8933)
- Umgang mit dem Ergebnisindikator auf MOVE (V7-9224)
- Verbessern Sie die Unterstützung des Schlüsselworts TEMPLATE für Feld- und Datenstruktur (V7-9278)
- Verbesserung von LIKEDS (mit LIKEDS definiertes DS wird automatisch qualifiziert) (V7-9302)
- COBOL — Verbesserung der Struktur der Generierung von Indikatoren (V7-9423)
- Der Const-Parameter im Prototyp ist nicht schreibgeschützt (V7-9437)
- Verbessern Sie das EDTCDE-Schlüsselwort mit dem Bearbeitungscode „Y“ (V7-9443)
- Support die Generierung des Feldes\*ROUTINE in PSDS und INFDS (V7-9487)
- Verbessern Sie das Umschreiben des Felds XXX in ein eigenständiges Feld (der Standardwert geht beim Umschreiben verloren) (V7-9522)
- Verbessern Sie die Support von DSPF-Schlüsselwörtern (V7-9658)
- Umgang mit dem ZEROES-Standardwert für Binärdateien (V7-9666)
- Support impliziter Zeiger (V7-9719)
- Verbessern Sie die Behandlung des integrierten Aufrufs %size mit einem Parameter (V7-9730)
- Verbessern Sie die Behandlung von Datenstrukturreferenzen in integrierten Aufrufen (%ELEM) (V7-9736)
- Verbessert den Umgang mit der Länge mit Vorzeichen für Felder mit LIKE-Referenz in der Definitionsspezifikation (V7-9738)
- Verbesserung gegenüber REWRITE (V7-9791)
- Verbesserung der Generierung von Indizes aus DDS-Dateien (V7-9803)
- Verbessern Sie die Robustheit von Mappern bei ungültigen numerischen Werten (V7-9813)

- Verbessern Sie die Generierung von SQLModel- und AllIndexes-Dateien (V7-9818)
- Verbessern Sie die qualifizierte DS-Unterstützung (V7-9863)
- Verbessern Sie die Unterstützung von LOOKUP (mit einem eigenständigen Feld WIE einem DS im Parameter) (V7-9961)
- Verbessern Sie LIKE am Indikator (V7-9985)
- Umgang mit dem resultierenden Indikator auf dem MVR (V7-9995)
- Support Zeichen N mit Tilde (V7-10021)
- Verbessern Sie die Generierung moderner DDL-Dateien aus älteren SQLDDS-Dateien (V7-10067)

## Transversale Fähigkeiten

### Neue Features

- Passen Sie den Ressourcenstandort mit einer yml-Eigenschaft an (D88816105)
- COBOL - Support der EXIT PERFORM-Anweisung zum Beenden einer Inline-PERFORM ohne Verwendung von GO TO/PERFORM... DURCH (V7-9582)
- Angabe der standardmäßigen Legacy-Kodierung, die in globalen Metadaten berücksichtigt werden soll. (V7-9883)

### Verbesserungen

- Verbessern Sie die Maskengenerierung (V7-9602)
- Verbessern Sie die Kontext-Aufwärmphase (V7-9621)
- Machen Sie den Charset CUSTOM930 threadsicher. (V7-9674)
- Verbesserung gegenüber MOVEA (V7-9773)

## Versionshinweise 3.5.0

Diese Version von AWS Blu Age Runtime and Modernization Tools bietet neue Funktionen sowohl für ältere zoS- als auch für AS400-Migrationen, die hauptsächlich auf die Optimierung von Datensätzen und Nachrichten ausgerichtet sind, sowie erweiterte Java-Funktionen als Ergebnis des Transformationsprozesses. Einige der wichtigsten Änderungen in dieser Version sind:

- Möglichkeit, CL-Programme zusätzlich zur bereits vorhandenen Groovy-Skriptfunktion nach Java zu migrieren, um die Integration mit anderen modernisierten Programmen zu erleichtern und die



Lernkurve für Kunden zu vereinfachen, indem die resultierende Programmiersprache vereinheitlicht wird.

- Zeitreduzierung und Optimierung der Leistung beim Laden von Datensätzen in Redis mit der neuen Datenbulk-Funktion.
- Fähigkeit, Datensätze innerhalb von Arbeitsschritten zu bearbeiten und weiterzugeben, um das Verhalten herkömmlicher Datensätze zu modernisieren.
- Erweiterung der SQL-Migration zur Unterstützung von VB-Eingabedateien und vereinfachter Java 11-Migration.
- Mehrere neue Mechanismen für eine schnellere Integration mit IBM MQ, darunter zusätzliche Header, erweiterte GET/PUT-Unterstützung und automatisches Abrufen von Warteschlangen-Metadaten.
- REST-Endpunkt für Datensatz-Metadaten und Import-Datensätze aus S3-Buckets.

Weitere Informationen zu den in dieser Version enthaltenen Änderungen finden Sie in den folgenden Abschnitten.

## Runtime-Version 3.5.0

### Themen

- [zOS](#)
- [ALS 400](#)
- [Transversale Fähigkeiten](#)

### zOS

#### Neue Features

- JCL SORT - Behandelt ein neues Schlüsselwort-Overlay (V7-9409)
- ZOS COBOL — erweiterte Unterstützung für schwebende Zeichen (V7-9404)
- Portierung von RedisJics TSQueue nach & (V7-9212) RedisTemplate ListOperations
- ZOS JCL - erweitert den Pfad des temporären Verzeichnisses um das Dateiverzeichnis, falls durch (V7-9012) definiert UserDefinedParameters
- Behandeln Sie FUNCTION ORD-MAX mit ALL (alle Array-Elemente) (V7-9366)

- Für Menschen lesbare Schlüssel mit Präfix werden jetzt beim Speichern von TS-Warteschlangen in Redis verwendet (V7-9212)
- Fügen Sie einen Get-Dataset-Endpunkt für die Blusam-API hinzu
- JCL — ADD-Unterstützung für Batch-Jobs mit Namen, die ein Sonderzeichen wie # enthalten (V7-9136)
- Das Abrufen von TSMODELS wird jetzt robust bei Bedarf durchgeführt (V7-9212)

## Verbesserungen

- Nicht versionierte INCLUDE-Unterstützung in LNK-Dateien (V7-6022)
- MQ - Verbesserte Kodierungsunterstützung (V7-9652)
- Verbesserung der Unterstützung für Doppelbytes oder gemischte Zeichensätze für unterschiedliche Zeichentypen (V7-9596)
- JCL - Support der FilesDirectory-Konfiguration in IDCAMS-Anweisungen zum Löschen von NONVSAM-Anweisungen (V7-9609)
- Support den Massenmodus für das Laden von ESDS- und RRDS-Datensätzen aus Dateien (V7-8639)
- Behandelt das Öffnen leerer ESDS im Eingabemodus. (V7-9287)
- Verbessern Sie die Anweisung DEFINE CLUSTER um die Unterstützung der Abkürzungen ORD/ UNORD (V7-9451)
- Leistungsverbesserungen bei Blusam Redis Lock (V7-8639)
- Erweiterung der Anweisung DEFINE CLUSTER zur Unterstützung von RECORDSIZE im Argumentbereich DATA () (V7-9337)
- Fügt die Unterstützung von BUFFERSPACE/UNIQUE-Attributen in DEFINE-CLUSTER-Anweisungen hinzu (V7-9419)
- Verbessert den Blusam-Lesevorgang für Datensatzdatensätze mit variabler Länge. (V7-9391)
- CICS-ADRESSE stellt fehlendes CWA korrekt als Null dar (V7-9491)
- Unnötiges Schreiben an Endsperrern entfernen (V7-8639)
- Behandeln Sie das Einfügen von Redis-Cache-Vorlagen in den Cache (V7-9510)
- Dekodieren Sie den BPXWDYN-Parameter korrekt (V7-9417)
- Verbesserung des LISTCAT-Exportverbrauchs (V7-9201)
- Unterstützung für nicht druckbare Zeichen im Namen von Blusam TS Queues (V7-9212)

- Steuert die Map-Erstellung für ein Feld mit Mapset Null (V7-9486)
- Verbessern BluesamRelativeFile Sie den Lösch- und Neuschreibvorgang für den dynamischen Zugriffsmodus. (V7-8989)

## ALS 400

### Neue Features

- Fügen Sie eine Funktion hinzu, um CL-Dateien als Java-Programme über einen standardmäßigen DS/STM-Pivot zu generieren (V7-9427)
- Support Eingabedateien mit ADD-Modus (V7-9378)
- Die Verwaltung der Sortierreihenfolge und des Abrufs wurde verbessert, um den CL-Befehl OPNQRYF (Open Query File) zu unterstützen, und die Unterstützung des SHARE-Parameters wurde hinzugefügt. OverrideItem (V7-9364)

### Verbesserungen

- Support von SFLNXTCHG auf (V7-8061) UpdateSubfile
- Ändern Sie den Bereich des CL-Kontextes, wenn Sie den CL-Befehl ausführen (V7-9624)
- Behandeln Sie den Rückgabecode für das Programm BPXWDYN (V7-9417)
- Löscht die lokalen Monitore. (V7-9624)
- Support des DSPF-Schlüsselworts RTNCSRLOC (V7-9389)
- set OnGreater OrEqual () setzt nicht gleich 1 (V7-9342)
- Aktualisieren Sie den Feld-Cache auf UpdateSubfileRecord (V7-9376)
- Verbessern Sie die Support SFLNXTCHG (V7-8061)

## Transversale Fähigkeiten

### Neue Features

- Ignorieren Sie das G-Präfix in der wörtlichen Grafikzeichenfolge. (V7-9420)
- ZOS COBOL - Verbesserte Unterstützung von Field.initialize () für einige spezielle Strukturen (V7-9485)
- Erlaubt die asynchrone Initialisierung des Kontextes, um die Leistung beim Programmstart zu verbessern (V7-9446)

- SQL veröffentlicht explizit die geöffnete Prepare-Anweisung und. ResultSet (V7-9422)
- Verbesserung von JMS MQ — Unterstützung von MQRFH2 für MQ PUT/V7-7085 — Unterstützung des Standard-Warteschlangen-Managers (V7-9400)
- SQL Management — Lambda-Konvertierungen für Parameter für SET-Befehle aktivieren (V7-9492)
- ZOS MQ JMS - Unterstützung für MQCOMIT und MQBACK hinzufügen (V7-9399)
- ZOS IBMQ — Verbesserte Unterstützung für MQINQ (V7-9544)
- Behandelt die CONCAT-Operation mit Byte statt mit Zeichenfolge, wenn Doppelbyte-Kodierung verwendet wird. (V7-8932)
- ZOS IBMMQ - Verbessern Sie die Unterstützung des PUT-Befehls mit den Optionen SET\_ALL\_CONTEXT (V7-9544)

### Verbesserungen

- Behandelt GDG-Dateinamen mit dem \$-Zeichen (V7-9066)
- Die SQL-Diagnose gibt 1 als NUMBER-Klausel zurück, wenn die vorherige SQL-Anweisung erfolgreich war. (V7-9410)
- Gliederung für ein Feld mit einer Länge ungleich Null (V7-7536)
- Support integrierte PL1-GRAFIK-Funktion (V7-9245)
- MQ — Unterstützung der Version für die Einstellung von MQGMO-Feldern hinzugefügt (V7-9500)
- JMS MQ GET - Die Nachricht hat eine Verbesserung der Datenlänge zurückgegeben (V7-9502)
- Setzt sqlerrd (3) mit der Anzahl der abgerufenen Elemente im ROWSET-Kontext. (V7-9371)

## Version 3.5.0 der Modernisierungstools

### Themen

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

### zOS

### Neue Features

- ZOS PLI - Support Sternchenindex bei Zuweisung mit binärem Ausdruck (V7-9178)
- JCL to BatchScript - Ein „//“ markiert das Ende der Jobausführung (V7-9304)
- ZOS PLI — verbessert die Unterstützung von Fließzeichen und Vorzeichen in numerisch bearbeitetem Typ (V7-8982)
- COBOL - Support der integrierten SUM-Funktion (V7-9367)
- JCL — optional, Kommentieren Sie toten Code nach einer Null-Anweisung (//) (V7-9202)
- JCL- Support des Operators '|' in der Zustandsanweisung (V7-9499)
- PL/I - Kommentar zu Direktiven zur Vorkompilierung im Vorverarbeitungsschritt, um das Analysieren von Ausnahmen zu verhindern (V7-9507)

### Verbesserungen

- Behandeln Sie die Stream-Definition mit einem Trennzeichen (V7-9615)
- Verbesserung der Handhabung von LISTCAT-Exporten. (V7-9201)
- PL/I — Erweiterung zur Unterstützung impliziter Null-Argumente (V7-9204)

## AS400

### Neue Features

- Support des DDS-Keywords CONCAT (V7-9439)
- Refaktorisieren Sie den generierten Java-Code für DSPF-Schlüsselwörter. (V7-7700)
- Support variierender Schlüsselwörter für Felder innerhalb einer Datenstrukturdefinition (V7-9029)

### Verbesserungen

- Verbessern Sie die Analyse der logischen Beziehung UND/ODER (V7-9352)
- COBOL Verbessert die Zuordnung zwischen vo und dsEntity (V7-9449)
- Leeren Wert anzeigen, wenn die numerische Eingabe fokussiert ist (V7-9374)
- Lokale Variable im SQL-Deklarationscursor (V7-9456)
- Bereichsproblem mit leerem DS (V7-9466)
- Kürzt Zeilen nach Spalte 80 vor dem Parsen (V7-9632)
- Verbessern Sie die Handhabung von Feldreferenzen und integrierten Aufrufen in Schlüsselwörtern (DIM, LIKE,...) in der Definitionsspezifikation (V7-9358)

- Support von SQL-Kommentaren (--) (V7-9632)
- FullFree Analysieren, geben Sie Date/Time/Timestamp (V7-9542) ein
- SQLCA aus FullFree der Analyse einbeziehen (V7-9333)
- Verbessern Sie die Support der Kontrollebene. (V7-9610)
- Bewältigen Sie den DS-Vergleich mit\*BLANKS (V7-9668)
- Verbessern Sie die Unterstützung mehrerer Indikatoren in DDS (V7-9318)
- Verbessern Sie die Unterstützung mehrerer DSPF-Programme (V7-9657)
- Verbessern Sie die Handhabung von Feldern mit LIKE (Fall von ähnlicher Datenstruktur und Fall von ähnlicher Datenstruktur in einem Array) (V7-9213)
- Kostenloses Rollenspiel, Fortsetzung buchstabengetreu (V7-9686)
- Verbessern Sie die Support von Aufzeichnungen zum Ende des Programms (V7-9452)
- Support der LINKAGE-Phrase in der CALL-Anweisung. (V7-9685)
- CASX-Operationscode (CASBB ohne CASX-Gruppe) (V7-9357)
- Verbessern Sie das RPG-Parsing (V7-9457) FullFree
- Das integrierte %LEN unterstützt DS nicht als Argument (V7-9267)
- Verbesserungen von MOVEA, wenn Faktor 2 \*ALL'X... 'ist (V7-9228)
- Support der Zuweisung mit dem Feld RENAME (V7-9385)

## Transversale Fähigkeiten

### Neue Features

- SQL Migrator-Tool — Fügen Sie die OID-Option für variable Datensatzlänge beim Laden von ebcDic hinzu. (V7-9380)
- SQL Migrator-Tool — Support für Java 11 auf OID-Option (V7-9599)

### Verbesserungen

- Verbessern Sie die Unterstützung für verschachtelte Arrays (V7-9595)
- Ersetze das Zeichen  $\hat{\rightarrow}$  durch! falls  $\hat{\rightarrow}$  von der Originalkodierung unterstützt wird. (V7-9465)
- JCL - Support der normalen PASS-Terminierung zur gemeinsamen Nutzung von Datensätzen zwischen Jobschritten (V7-9504)

- Wenden Sie ON NULL auf die Spaltendefinition in ORACLE an, wenn es um VARCHAR und den DB-Spaltentyp geht, der Nullwerte zulässt. (V7-9681)
- Verbessern Sie die Konformität mit der Federeinspritzung (V7-9635)

## AWS Anweisungen zum Upgrade auf Blu Age

Diese Seite enthält Anweisungen zum Upgrade der AWS Blu Age-Version.

### Migration von 3.10.0 auf 4.0.0

Die wichtigste Änderung in 4.0.0 ist die Migration von Spring Boot 2.7 auf Spring Boot 3.2 und von Tomcat 9 auf Tomcat 10.

### Änderungen am Code

In diesem Abschnitt werden die Änderungen aufgeführt, die erforderlich sind, um den modernisierten Code mit AWS Blu Age Runtime 4.0.0 kompatibel zu machen. Sie können diesen Abschnitt überspringen, wenn Sie sich entscheiden, eine neue Generation mit der Version 4.0.0 auf Blu Insights (Transformation Center) auf den Markt zu bringen.

### POM-Änderungen

Gruppe	ArtifactId	Änderung
org.slf4j	slf4j-api	Entfernen (ist eine transitive Abhängigkeit)
org.yaml	Snakeyaml	Entfernen (ist eine transitive Abhängigkeit)
org.springframework.boot	spring-boot-starter-web	- Aktualisieren Sie spring.boot.version auf 3.2.4 — Entfernen Sie den Ausschluss von log4j-to-slf
org.springframework.boot	spring-boot-starter-jta-Atomikos	Wechseln Sie zu com.atomikos:3-starter:6.0.0 transactions-spring-boot

Gruppe	ArtifactId	Änderung
org.apache.commons	commons-dbcp2	Führen Sie ein Upgrade auf 2.10.0 durch
org.postgresql	postgresql	Führen Sie ein Upgrade auf 42.7.2 durch
com.microsoft.sqlserver	mssql-jdbc	Führen Sie ein Upgrade auf 12.4.2.jre11 durch
com.oracle.database.jdbc	ojdbc 8	Wechseln Sie zu ojdbc11 Version 23.3.0.23.09

### Migrieren Sie von Javax nach Jakarta

Das Tomcat-Upgrade beinhaltet eine Migration vom Javax-Java-Paket nach Jakarta. Stellen Sie sicher, dass Sie Ihre Importe von `javax.*` nach Jakarta entsprechend aktualisieren. \*

Fast alle alten Klassen, auf die im Javax-Paket verwiesen wurde, befinden sich in Jakarta. Bekannte Ausnahmen sind die `javax.xml` Pakete `javax.sql` und, die immer noch unverändert sind.

### Atomikos ändern sich

Aufgrund der oben genannten Änderung der Abhängigkeit

`org.springframework.boot.jta.atomikos.AtomikosDataSourceBean` müssen Verweise auf geändert werden in `com.atomikos.spring.AtomikosDataSourceBean`

### Entfernung des PostgreSQL-Dialekts

Die benutzerdefinierte Klasse `PostgreSQLDialect.java` wird entfernt. Verweise darauf im Haupt-Launcher müssen ebenfalls entfernt werden.

### Bereitstellung (AWS Blu Age Runtime (nicht verwaltet))

#### Tomcat

Diese Version ist mit 10.1.17 Tomcat kompatibel. Für die Ausführung der Blu Age Runtime ist ein Upgrade des Tomcat-Servers auf diese Version erforderlich. 4.0.0 Stellen Sie sicher, dass Sie die alten Konfigurationsänderungen (insbesondere die Catalina-Eigenschaften) portieren.



## Geteilte Abhängigkeiten

Der gemeinsam genutzte Runtime-Ordner enthält die up-to-date Abhängigkeiten.

## Zusätzliche Abhängigkeiten

Wenn Sie zusätzliche Abhängigkeiten verwendet haben (die nicht in der Laufzeit enthalten sind), müssen Sie diese möglicherweise aktualisieren. In der Readme-Datei im zusätzlichen Ordner sind die unterstützten Versionen aufgeführt.

# AWS Blu Age Runtime-Konzepte

Wenn Sie die grundlegenden Konzepte der AWS Blu Age Runtime verstehen, können Sie besser verstehen, wie Ihre Anwendungen durch automatisiertes Refactoring modernisiert werden.

## Themen

- [AWS Hochrangige Architektur von Blu Age Runtime](#)
- [AWS Blu-Age-Struktur einer modernisierten Anwendung](#)
- [Datenvereinfacher](#)

## AWS Hochrangige Architektur von Blu Age Runtime

Als Teil der AWS Blu Age-Lösung für die Modernisierung älterer Programme auf Java bietet die AWS Blu Age Runtime einen einheitlichen, REST-basierten Einstiegspunkt für modernisierte Anwendungen und ein Framework für die Ausführung solcher Anwendungen, indem Bibliotheken ältere Konstrukte bereitstellen und die Organisation des Programmcodes standardisiert wird.

Solche modernisierten Anwendungen sind das Ergebnis des AWS Blu Age Automated Refactor-Prozesses zur Modernisierung von Mainframe- und Midrange-Programmen (im folgenden Dokument als „Legacy“ bezeichnet) auf eine webbasierte Architektur.

Die Ziele von AWS Blu Age Runtime sind die Reproduktion des Verhaltens älterer Programme (Isokompatibilität), die Leistung (in Bezug auf die Ausführungszeit und den Ressourcenverbrauch) und die einfache Wartung modernisierter Programme durch Java-Entwickler unter Verwendung vertrauter Umgebungen und Redewendungen wie Tomcat, Spring, Getter/Setter und Fluent APIs.

## Themen

- [AWS Blu Age Runtime-Komponenten](#)

- [Ausführungsumgebungen](#)
- [Staatenlosigkeit und Umgang mit Sitzungen](#)
- [Hohe Verfügbarkeit und Staatenlosigkeit](#)

## AWS Blu Age Runtime-Komponenten

Die AWS Blu Age Runtime-Umgebung besteht aus zwei Arten von Komponenten:

- Eine Reihe von Java-Bibliotheken (Jar-Dateien), die häufig als „gemeinsam genutzter Ordner“ bezeichnet werden und ältere Konstrukte und Anweisungen bereitstellen.
- Eine Reihe von Webanwendungen (WAR-Dateien), die auf Spring basierende Webanwendungen enthalten und einen gemeinsamen Satz von Frameworks und Diensten für modernisierte Programme bereitstellen.

In den folgenden Abschnitten wird die Rolle dieser beiden Komponenten detailliert beschrieben.

### AWS Blue-Age-Bibliotheken

Bei den AWS Blu Age-Bibliotheken handelt es sich um eine Reihe von JAR-Dateien, die in einem `shared/` Unterordner gespeichert sind, der dem Standard-Tomcat-Klassenpfad hinzugefügt wurde, um sie allen modernisierten Java-Programmen zur Verfügung zu stellen. Ihr Ziel ist es, Funktionen bereitzustellen, die in der Java-Programmierungsumgebung weder nativ noch einfach verfügbar sind, aber für ältere Entwicklungsumgebungen typisch sind. Diese Funktionen werden auf eine Weise bereitgestellt, die Java-Entwicklern so vertraut wie möglich ist (Getter/Setter, klassenbasierte, fließende APIs). Ein wichtiges Beispiel ist die Data Simplifier-Bibliothek, die ältere Speicherlayout- und Manipulationskonstrukte (wie sie in den Sprachen COBOL, PL1 oder RPG vorkommen) für Java-Programme bereitstellt. Diese JAR-Dateien sind eine Kernabhängigkeit des modernisierten Java-Codes, der aus älteren Programmen generiert wurde. Weitere Informationen zum Data Simplifier finden Sie unter [Datenvereinfacher](#)

### Webanwendung

Web Application Archives (WARs) sind eine Standardmethode für die Bereitstellung von Code und Anwendungen auf dem Tomcat-Anwendungsserver. Die im Rahmen der AWS Blu-Age-Laufzeit bereitgestellten Frameworks zielen darauf ab, eine Reihe von Ausführungs-Frameworks bereitzustellen, die ältere Umgebungen und Transaktionsmonitore (JCL-Batches, CICS, IMS...) und die zugehörigen erforderlichen Dienste reproduzieren.

Der wichtigste ist `gapwalk-application` (oft abgekürzt als „Gapwalk“), der einen einheitlichen Satz von REST-basierten Einstiegspunkten zur Auslösung und Steuerung von Transaktionen, Programmen und Batch-Ausführung bietet. Weitere Informationen finden Sie unter [AWS Blu-Age-Laufzeit-APIs](#).

Diese Webanwendung weist Java-Ausführungsthreads und Ressourcen zu, um modernisierte Programme in dem Kontext auszuführen, für den sie entworfen wurden. Beispiele für solche reproduzierten Umgebungen werden im folgenden Abschnitt detailliert beschrieben.

Andere Webanwendungen fügen der Ausführungsumgebung (genauer gesagt der unten beschriebenen „Programmregistrierung“) Programme hinzu, die diejenigen emulieren, die für die älteren Programme verfügbar sind und von diesen aufgerufen werden können. Zwei wichtige Kategorien davon sind:

- Emulation von vom Betriebssystem bereitgestellten Programmen: Insbesondere JCL-gesteuerte Batches erwarten in ihrer Standardumgebung die Fähigkeit, eine Vielzahl von Programmen zur Bearbeitung von Dateien und Datenbanken aufrufen zu können. SORTDFSORTIDCAMS Zu den Beispielen gehören/oder. Zu diesem Zweck werden Java-Programme bereitgestellt, die ein solches Verhalten reproduzieren und mit den gleichen Konventionen wie die älteren Programme aufgerufen werden können.
- „Treiber“, bei denen es sich um spezialisierte Programme handelt, die vom Ausführungsframework oder der Middleware als Einstiegspunkte bereitgestellt werden. Ein Beispiel ist CBLTDLI, auf welche COBOL-Programme, die in der IMS-Umgebung ausgeführt werden, angewiesen sind, um auf IMS-bezogene Dienste (IMS-Datenbank, Benutzerdialog über MFS usw.) zuzugreifen.

## Registrierung der Programme

Um an diesen Konstrukten, Frameworks und Diensten teilzuhaben und diese zu nutzen, folgen Java-Programme, die aus älteren Versionen modernisiert wurden, einer bestimmten Struktur, die unter dokumentiert ist. [AWS Blu-Age-Struktur einer modernisierten Anwendung](#) Beim Start sammelt die AWS Blu Age Runtime all diese Programme in einer gemeinsamen „Programmregistrierung“, sodass sie anschließend aufgerufen (und sich gegenseitig aufgerufen) werden können. Das Program Registry bietet lockere Kopplungen und Möglichkeiten der Zerlegung (da Programme, die sich gegenseitig aufrufen, nicht gleichzeitig modernisiert werden müssen).

## Ausführungsumgebungen

Häufig anzutreffende ältere Umgebungen und Choreografien sind verfügbar:

- JCL-gesteuerte Batches können nach der Modernisierung auf Java-Programme und Groovy-Skripte synchron (blockierend) oder asynchron (getrennt) gestartet werden. Im letzteren Fall kann ihre Ausführung über REST-Endpunkte überwacht werden.
- Ein AWS Blu Age-Subsystem bietet eine Ausführungsumgebung, die CICS ähnelt, und zwar durch:
  - einen Einstiegspunkt, der zum Starten einer CICS-Transaktion und zum Ausführen der zugehörigen Programme verwendet wird, wobei die CICS-Choreographie der „Run-Levels“ eingehalten wird,
  - ein externer Speicher für Ressourcendefinitionen,
  - ein homogener Satz fließender Java-APIs, die Anweisungen reproduzieren EXEC CICS,
  - eine Reihe von austauschbaren Klassen, die CICS-Dienste wie Temporary Storage Queues, Temporary Data Queues oder Dateizugriff reproduzieren (in der Regel sind mehrere Implementierungen verfügbar, wie Amazon Managed Service für Apache Flink, Amazon Simple Queue Service oder RabbitMQ für TD Queues),
  - Für benutzerorientierte Anwendungen wurde das BMS-Bildschirmbeschreibungsformat zu einer Angular-Webanwendung modernisiert, und der entsprechende „Pseudo-Konversationsdialog“ wird unterstützt.
- In ähnlicher Weise bietet ein anderes Subsystem eine auf IMS-Nachrichten basierende Choreographie und unterstützt die Modernisierung von Benutzeroberflächenbildschirmen im MFS-Format.
- Darüber hinaus ermöglicht ein drittes Subsystem die Ausführung von Programmen in einer iSeries-ähnlichen Umgebung, einschließlich der Modernisierung von DSPF-spezifischen (Display File) -spezifischen Bildschirmen.

All diese Umgebungen bauen auf gängigen Diensten auf Betriebssystemebene auf, wie z. B.:

- die Emulation herkömmlicher Speicherzuweisungen und Layouts (Data Simplifier),
- Thread-basierte Reproduktion der COBOL-Ausführung von „Run-Units“ und des Mechanismus zur Übergabe von Parametern (Anweisung), CALL
- Emulation von flachen, verketteten, VSAM-Organisationen (mithilfe der Blusam-Bibliotheken) und GDG-Datensatzorganisationen,
- Zugriff auf Datenspeicher wie RDBMS (Statements). EXEC SQL

## Staatenlosigkeit und Umgang mit Sitzungen

Ein wichtiges Merkmal der AWS Blu Age Runtime besteht darin, Hochverfügbarkeits- (HA) und horizontale Skalierbarkeitsszenarien bei der Ausführung modernisierter Programme zu ermöglichen.

Der Eckpfeiler dafür ist Staatenlosigkeit. Ein wichtiges Beispiel dafür ist die Behandlung von HTTP-Sitzungen.

### Behandlung von Sitzungen

Da Tomcat webbasiert ist, sind die HTTP-Sitzungsbehandlung (wie sie von Tomcat und Spring bereitgestellt wird) und das stateless Design ein wichtiger Mechanismus dafür. Daher basiert das Konzept der Staatenlosigkeit auf den folgenden Grundsätzen:

- Benutzer stellen eine Verbindung über HTTPS her,
- Anwendungsserver werden hinter einem Load Balancer bereitgestellt,
- Wenn ein Benutzer zum ersten Mal eine Verbindung zur Anwendung herstellt, wird diese authentifiziert und der Anwendungsserver erstellt eine Kennung (normalerweise in einem Cookie)
- Diese Kennung wird als Schlüssel zum Speichern und Abrufen des Benutzerkontextes in oder aus einem externen Cache (Datenspeicher) verwendet.

Die Cookieverwaltung erfolgt automatisch durch das AWS Blu Age-Framework und den zugrunde liegenden Tomcat-Server. Dies ist für den Benutzer transparent. Der Internetbrowser des Benutzers verwaltet dies automatisch.

Die Gapwalk-Webanwendung kann den Sitzungsstatus (den Kontext) in verschiedenen Datenspeichern speichern:

- Amazon ElastiCache für Redis
- Redis-Cluster
- in der Speicherzuweisung (nur für Entwicklungs- und Standalone-Umgebungen, nicht für HA geeignet).

## Hohe Verfügbarkeit und Staatenlosigkeit

Allgemeiner ausgedrückt ist ein Grundprinzip des AWS Blu Age-Frameworks Staatenlosigkeit: Die meisten nichttransienten Zustände, die zur Reproduktion des Verhaltens älterer Programme

erforderlich sind, werden nicht auf den Anwendungsservern gespeichert, sondern über eine externe, gemeinsame „zentrale Informationsquelle“ gemeinsam genutzt.

Beispiele für solche Zustände sind die temporären Speicherwarteschlangen oder Ressourcendefinitionen von CICS, und typische externe Speicher für diese sind Redis-kompatible Server oder relationale Datenbanken.

Dieses Design führt in Kombination mit Lastenausgleich und gemeinsam genutzten Sitzungen dazu, dass die meisten benutzerseitigen Dialoge (OLTP, „Online Transactional Processing“) auf mehrere „Knoten“ (hier Tomcat-Instanzen) verteilt werden können.

In der Tat kann ein Benutzer eine Transaktion auf einem beliebigen Server ausführen und es ist ihm egal, ob der nächste Transaktionsaufruf auf einem anderen Server ausgeführt wird. Wenn dann ein neuer Server gestartet wird (aufgrund von Auto Scaling oder um einen fehlerhaften Server zu ersetzen), können wir garantieren, dass jeder erreichbare und fehlerfreie Server die Transaktion wie erwartet mit den richtigen Ergebnissen ausführen kann (erwarteter Rückgabewert, erwartete Datenänderung in der Datenbank usw.).

## AWS Blu-Age-Struktur einer modernisierten Anwendung

Dieses Dokument enthält Einzelheiten zur Struktur modernisierter Anwendungen (mithilfe der Refactoring-Tools für die AWS Mainframe-Modernisierung), sodass Entwickler verschiedene Aufgaben ausführen können, z. B.:

- reibungsloses Navigieren in Anwendungen.
- Entwicklung von kundenspezifischen Programmen, die von den modernisierten Anwendungen aus aufgerufen werden können.
- sicheres Refactoring modernisierter Anwendungen.

Wir gehen davon aus, dass Sie bereits über Grundkenntnisse in den folgenden Bereichen verfügen:

- ältere gängige Codierungskonzepte wie Datensätze, Datensätze und ihre Zugriffsmodi auf Datensätze (indexiert, sequentiell), VSAM, Run-Units, JCL-Skripte, CICS-Konzepte usw.
- [Java-Codierung mit dem Spring-Framework](#).
- Im gesamten Dokument verwenden wir aus short class names Gründen der Lesbarkeit. Weitere Informationen finden Sie unter [AWS Vollqualifizierte Namenszuordnungen von Blu Age](#). So rufen Sie die entsprechenden vollqualifizierten Namen für AWS Blu-Age-Laufzeitelemente ab

und [Vollqualifizierte Namenszuordnungen von Drittanbietern](#) So rufen Sie die entsprechenden vollqualifizierten Namen für Elemente von Drittanbietern ab.

- [Alle Artefakte und Proben stammen aus den Ergebnissen des Modernisierungsprozesses der CardDemo COBOL/CICS-Beispielanwendung.](#)

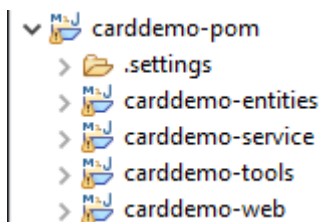
## Themen

- [Organisation der Artefakte](#)
- [Programme ausführen und aufrufen](#)
- [Schreiben Sie Ihr eigenes Programm](#)
- [Vollständig qualifizierte Namenszuordnungen](#)

## Organisation der Artefakte

AWS Die modernisierten Anwendungen von Blu Age sind als Java-Webanwendungen (.war) verpackt, die Sie auf einem JEE-Server bereitstellen können. In der Regel handelt es sich bei dem Server um eine [Tomcat-Instanz](#), die die AWS Blu Age Runtime einbettet, die derzeit auf den Frameworks [Springboot](#) und [Angular](#) (für die Benutzeroberfläche) aufbaut.

Der Krieg aggregiert mehrere Komponentenartefakte (.jar). Jedes JAR ist das Ergebnis der Kompilierung (mit dem [Maven-Tool](#)) eines speziellen Java-Projekts, dessen Elemente das Ergebnis des Modernisierungsprozesses sind.



Die grundlegende Organisation basiert auf der folgenden Struktur:

- Entitätenprojekt: enthält Geschäftsmodell- und Kontextelemente. Der Projektname endet in der Regel mit „-entities“. In der Regel entspricht dies für ein bestimmtes COBOL-Legacy-Programm der Modernisierung des I/O-Bereichs (Datensätze) und der Datenaufteilung. Sie können mehr als ein Entitätsprojekt haben.
- Serviceprojekt: enthält ältere Elemente zur Modernisierung der Geschäftslogik. In der Regel die Prozedurabteilung eines COBOL-Programms. Sie können mehr als ein Serviceprojekt haben.

- **Hilfsprojekt:** enthält gemeinsam genutzte Tools und Dienstprogramme, die von anderen Projekten verwendet werden.
- **Webprojekt:** beinhaltet gegebenenfalls die Modernisierung von Elementen im Zusammenhang mit der Benutzeroberfläche. Wird nicht für reine Batch-Modernisierungsprojekte verwendet. Diese Benutzeroberflächenelemente könnten aus CICS-BMS-Maps, IMS-MFS-Komponenten und anderen Mainframe-UI-Quellen stammen. Sie können mehr als ein Webprojekt haben.

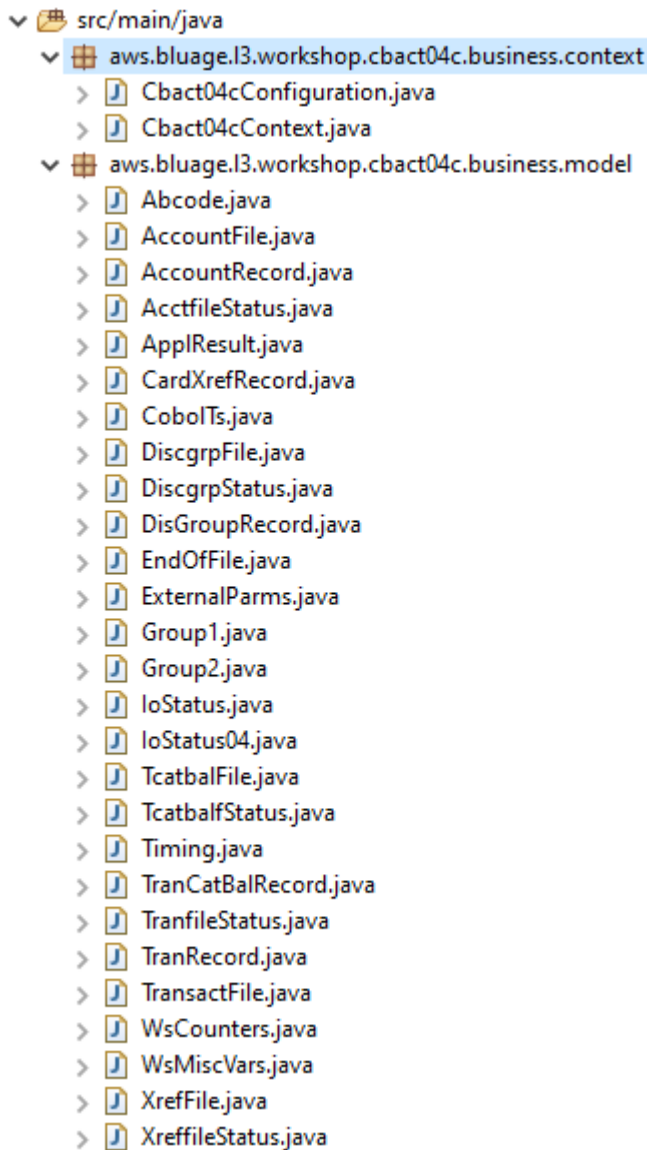
## Entitäten, Projektinhalte

### Note

Die folgenden Beschreibungen gelten nur für COBOL- und PL/I-Modernisierungsausgaben. Die Ausgaben für die RPG-Modernisierung basieren auf einem anderen Layout.

Vor jedem Refactoring ist die Paketorganisation im Entitätsprojekt an die modernisierten Programme gebunden. Sie können dies auf verschiedene Arten erreichen. Die bevorzugte Methode ist die Verwendung der Refactoring-Toolbox, die aktiv ist, bevor Sie den Codegenerierungsmechanismus auslösen. Dies ist eine fortgeschrittene Operation, die in den BluAge Schulungen erklärt wird. Weitere Informationen finden Sie unter [Refactoring-Workshop](#). Dieser Ansatz ermöglicht es Ihnen, den Java-Code später erneut zu generieren, um beispielsweise in future von weiteren Verbesserungen zu profitieren). Die andere Möglichkeit besteht darin, regelmäßiges Java-Refactoring direkt am generierten Quellcode durchzuführen und dabei einen beliebigen Java-Refactoring-Ansatz zu verwenden, den Sie möglicherweise anwenden möchten — auf eigenes Risiko.





## Kurse im Zusammenhang mit dem Programm

Jedes modernisierte Programm besteht aus zwei Paketen, einem business.context- und einem business.model-Paket.

- *base package.program.business.context*

Das Unterpaket business.context enthält zwei Klassen, eine Konfigurationsklasse und eine Kontextklasse.

- Eine Konfigurationsklasse für das Programm, die spezifische Konfigurationsdetails für das jeweilige Programm enthält, z. B. den Zeichensatz für die Darstellung zeichenbasierter Datenelemente, den Standard-Bytewert für das Auffüllen von

Datenstrukturelementen usw. Der Klassenname endet mit „Configuration“. Er ist mit der `@org.springframework.context.annotation.Configuration` Anmerkung gekennzeichnet und enthält eine einzige Methode, die ein korrekt eingerichtetes `Configuration` Objekt zurückgeben muss.

```
Cbact04cConfiguration.java ×
1 package aws.bluage.13.workshop.cbact04c.business.context;
2
3 import com.netfactive.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
```

- Eine Kontextklasse, die als Brücke zwischen den Dienstklassen des Programms (siehe unten) und den Datenstrukturen (Record) und Datensätzen (File) aus dem Modell-Unterpaket (siehe unten) dient. Der Klassenname endet mit „Context“ und ist eine Unterklasse der `RuntimeContext` Klasse.

```

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

```

- *base package.program.business.model*

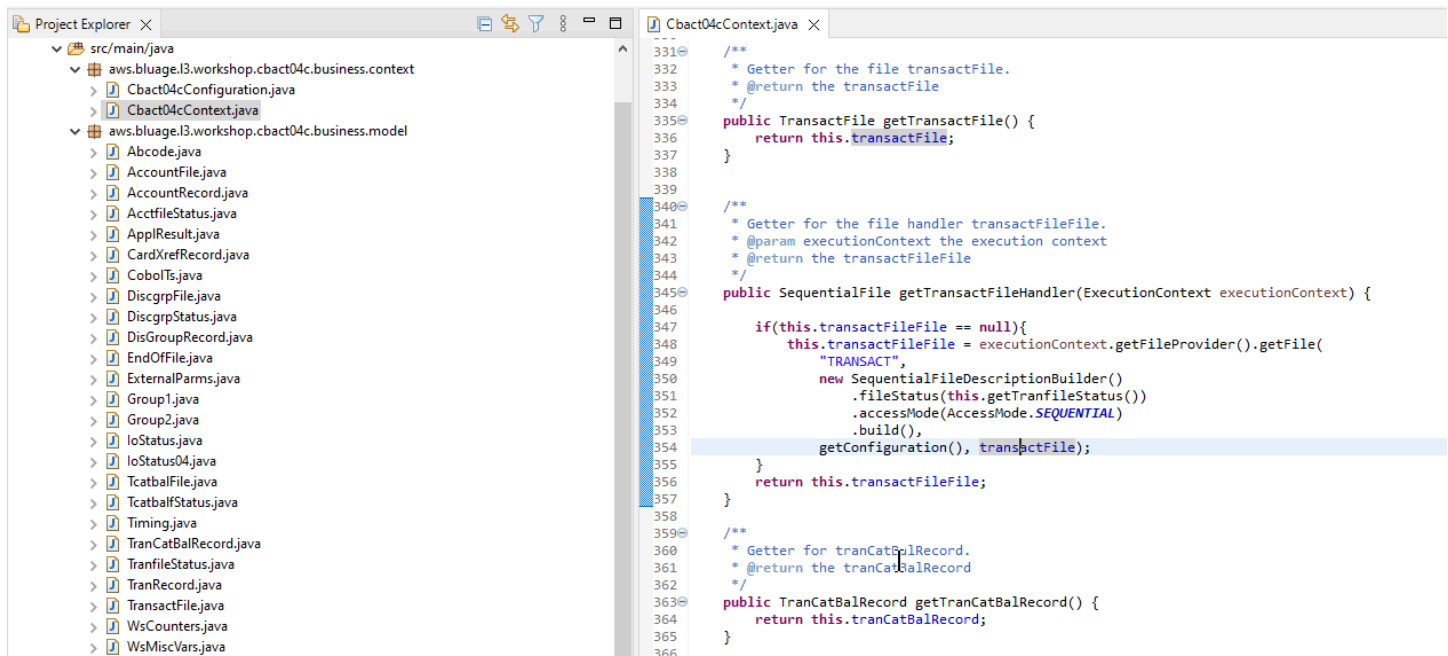
Das Modell-Unterpaket enthält alle Datenstrukturen, die das angegebene Programm verwenden kann. Beispielsweise entspricht jede COBOL-Datenstruktur der Ebene 01 einer Klasse im Modellunterpaket (Datenstrukturen auf niedrigerer Ebene sind Eigenschaften ihrer eigenen Struktur der Ebene 01). Weitere Informationen darüber, wie wir 01-Datenstrukturen modernisieren, finden Sie unter. [Datenvereinfacher](#)

```

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

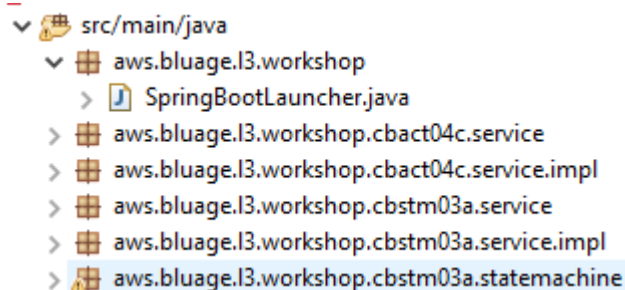
```

Alle Klassen erweitern die RecordEntity Klasse, die den Zugriff auf eine Geschäftsdatensatzdarstellung darstellt. Einige der Datensätze haben einen speziellen Zweck, da sie an eine gebunden sind File. Die Bindung zwischen a Record und a File erfolgt in den entsprechenden FileHandler \*-Methoden, die sich bei der Erstellung des Dateiobjekts in der Kontextklasse befinden. Die folgende Liste zeigt beispielsweise, wie das an TransactfileFile File das TransactFile Record (aus dem Modell-Unterpaket) gebunden ist.



## Inhalt des Serviceprojekts

Jedes Serviceprojekt wird mit einer speziellen [Springboot-Anwendung](#) geliefert, die als Rückgrat der Architektur verwendet wird. Dies wird durch die angegebene Klasse materialisiert `SpringBootLauncher`, die sich im Basispaket der Java-Quellen des Dienstes befindet:



Diese Klasse ist insbesondere verantwortlich für:

- stellt die Verbindung zwischen den Programmklassen und den verwalteten Ressourcen her (Datenquellen/ Transaktionsmanager/ Mappings von Datensätzen/etc...).
- Bereitstellung von zwei Programmen. `ConfigurableApplicationContext`
- entdeckt alle Klassen, die als Spring-Komponenten (`@Component`) gekennzeichnet sind.
- Sicherstellen, dass die Programme ordnungsgemäß in der `ProgramRegistry` Datei registriert sind -- siehe die für diese Registrierung zuständige Initialisierungsmethode.

```
/**
 * 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);
}
```

## Programmbezogene Artefakte

Ohne vorheriges Refactoring werden die Ergebnisse der Modernisierung der Geschäftslogik auf Basis von zwei oder drei Paketen pro Legacy-Programm organisiert:

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

Der umfassendste Fall umfasst drei Pakete:

- `base package.program.service`: enthält eine Schnittstelle namens `Program Process`, die über Geschäftsmethoden zur Verwaltung der Geschäftslogik verfügt, wobei der bisherige Ausführungskontrollfluss erhalten bleibt.
- `base package.program.service.impl`: enthält eine Klasse namens `ProgramProcessImpl`, bei der es sich um die Implementierung der zuvor beschriebenen `Process`-Schnittstelle handelt. Hier werden die alten Anweisungen in Java-Anweisungen „übersetzt“, wobei auf das AWS Blu Age-Framework zurückgegriffen wird:

```

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

```

- `base package.program.statemachine`: Dieses Paket ist möglicherweise nicht immer vorhanden. Es ist erforderlich, wenn bei der Modernisierung der alten Ablaufsteuerung ein State-Machine-Ansatz (d. h. die Verwendung des [State Machine Spring-Frameworks](#)) verwendet werden muss, um den alten Ausführungsablauf ordnungsgemäß abzudecken.



In diesem Fall enthält das StateMachine-Unterpaket zwei Klassen:

- **ProgramProcedureDivisionStateMachineController**: eine Klasse, die eine Klasse erweitert, die die Schnittstellen `StateMachineController` (definiert Operationen, die zur Steuerung der Ausführung einer Zustandsmaschine erforderlich sind) und `StateMachineRunner` (definiert Operationen, die für den Betrieb einer Zustandsmaschine erforderlich sind) implementiert, die zur Steuerung der Spring State Machine-Mechanik verwendet werden; zum Beispiel die, `SimpleStateMachineController` wie im Beispielfall.

```

1 package aws.bluage.l3.workshop.cocrdslc.statemachine;
2
3 import aws.bluage.l3.workshop.cocrdslc.business.context.CocrdslcContext;
4
5 /**
6  * Controller managing the state machine "CocrdslcProcedureDivisionStateMachine" execution.
7  */
8 @Component("aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineController")
9 @Import({
10     aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineService.class
11 })
12 @Lazy
13 public class CocrdslcProcedureDivisionStateMachineController extends SimpleStateMachineController<States, Events> {
14
15     /**
16      * State machine states.
17      */
18     public enum States {
19         _0000_MAIN_1, _0000_MAIN, ABEND_ROUTINE, FINAL, LOCAL_FINAL
20     }
21
22     /**
23      * State machine events.
24      */
25     public enum Events {
26         TO_0000_MAIN_1, TO_0000_MAIN, TO_ABEND_ROUTINE, TO_FINAL, TO_LOCAL_FINAL
27     }
28
29     /**
30      * State machine state process service provider.
31      */
32     @Autowired
33     @Lazy
34     private CocrdslcProcedureDivisionStateMachineService stateProcess;
35
36     @Override
37     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
38         throw new UnsupportedOperationException("Please use the four arguments configureStateMachine method instead: configureStateMachine(StateMachineStateConfigurer<States, Events> states, "
39             + "StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl)");
40     }
41
42     @Override
43     protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl) throws Exception {
44         StateConfigurer<States, Events> configurator = states.withStates();
45         configurator.initial(States._0000_MAIN_1).end(States.FINAL);
46         configurator.state(States._0000_MAIN_1);
47         configurator.state(States.FINAL);
48
49         StateConfigurer<States, Events> subConfigurator = states.withStates().parent(States._0000_MAIN_1);
50         subConfigurator.initial(States._0000_MAIN).end(States.LOCAL_FINAL);
51         CocrdslcContext lctx = (CocrdslcContext) ctx;
52         subConfigurator.state(States._0000_MAIN, buildAction(() -> {stateProcess.configureMain(lctx, ctrl)}), null);
53         subConfigurator.state(States.ABEND_ROUTINE, buildAction(() -> {stateProcess.abendRoutine(lctx, ctrl)}), null);
54
55         configureTransitions(transitions);
56     }
57
58     /**
59      * Declare state machine transitions.
60      * @param transitions the transitions configuration helper
61      */
62     private void configureTransitions(StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
63         transitions.withLocal().source(States._0000_MAIN_1).target(States.ABEND_ROUTINE).event(Events.TO_ABEND_ROUTINE);
64         transitions.withExternal().source(States.ABEND_ROUTINE).target(States.FINAL).event(Events.TO_FINAL);
65     }
66 }
67
68
69
70
71
72
73
74
75
76
77
78
79
80

```

Der State-Machine-Controller definiert die möglichen unterschiedlichen Zustände und die Übergänge zwischen ihnen, wodurch der bisherige Ausführungssteuerungsablauf für das jeweilige Programm wiedergegeben wird.

Beim Erstellen der Zustandsmaschine bezieht sich der Controller auf Methoden, die in der zugehörigen Serviceklasse definiert sind, die sich im State-Machine-Paket befindet und im Folgenden beschrieben wird:

```

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

```

- *ProgramProcedureDivisionStateMachineService*: Diese Serviceklasse stellt eine gewisse Geschäftslogik dar, die an die Zustandsmaschine gebunden werden muss, die der Zustandsmaschine-Controller erstellt, wie zuvor beschrieben.

Der Code in den Methoden dieser Klasse verwendet die Ereignisse, die im State-Machine-Controller definiert sind:

```

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

```

```

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

```

Der StateMachine-Dienst ruft auch die zuvor beschriebene Implementierung des Prozessdienstes auf:

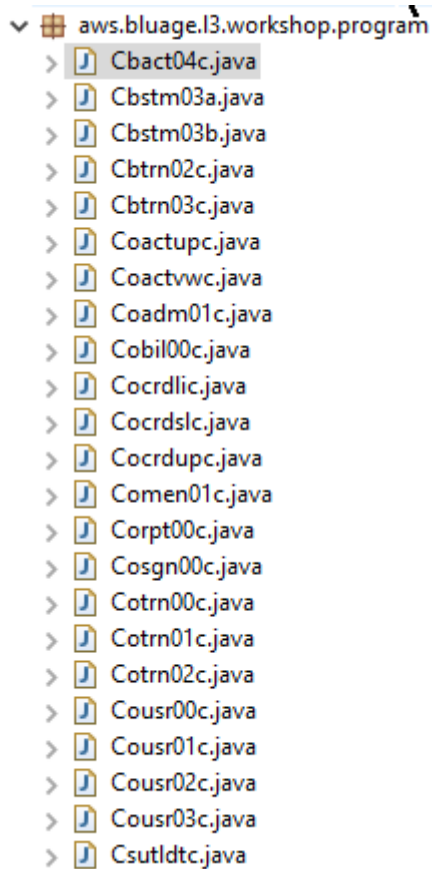
```

CocrdslcProcedureDivisionStateMachineService.java X
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.getChworkAreas().setCcacctIdN(ctx.getCarddemoCommarea().getCdemoAcctId());
175      ctx.getChworkAreas().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.getChworkAreas().setCardErrorMsg(ctx.getWsMiscStorage().getWsReturnMsg());
211      instanceCocrdslcProcess.sendMap(ctx, ctrl);
212      instanceCocrdslcProcess.commonReturn(ctx, ctrl);
213  }
214  instanceCocrdslcProcess.commonReturn(ctx, ctrl);
215

```

Darüber hinaus *base package*.program spielt ein benanntes Paket eine wichtige Rolle, da es eine Klasse pro Programm zusammenfasst, die als Einstiegspunkt für das Programm dient

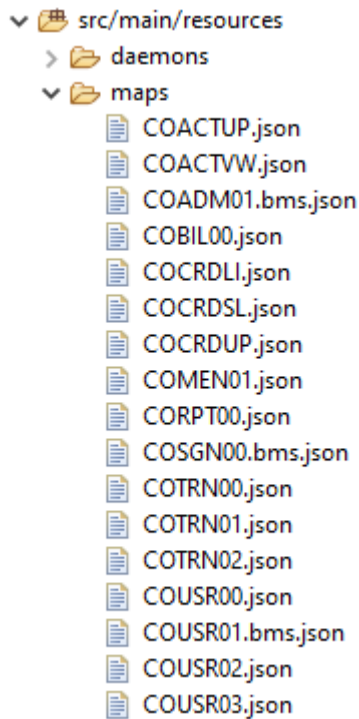
(mehr dazu später). Jede Klasse implementiert die Program Schnittstelle, die Markierung für einen Programmeinstiegspunkt.



## Andere Artefakte

- Begleiter von BMS MapS

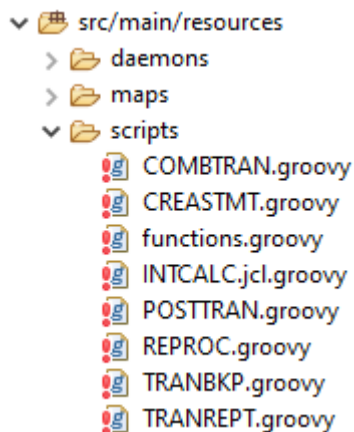
Neben programmbezogenen Artefakten kann das Dienstprojekt auch andere Artefakte für verschiedene Zwecke enthalten. Bei der Modernisierung einer CICS-Online-Anwendung wird im Rahmen des Modernisierungsprozesses eine JSON-Datei erstellt und im Map-Ordner des Ordners `/src/main/resources` abgelegt:



Die Blu Age-Laufzeit verwendet diese JSON-Dateien, um die von der SEND MAP-Anweisung verwendeten Datensätze mit den Bildschirmfeldern zu verknüpfen.

- Groovy-Skripte

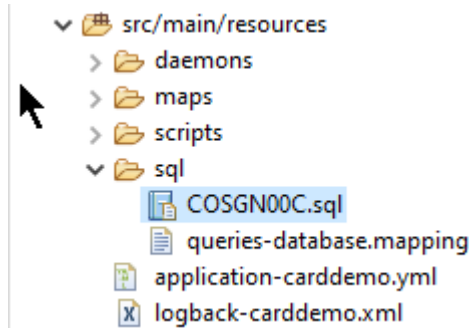
Wenn die alte Anwendung über JCL-Skripte verfügte, wurden diese als [Groovy-Skripte](#) modernisiert und im Ordner `/src/main/resources/scripts` gespeichert (mehr zu diesem speziellen Ort später):



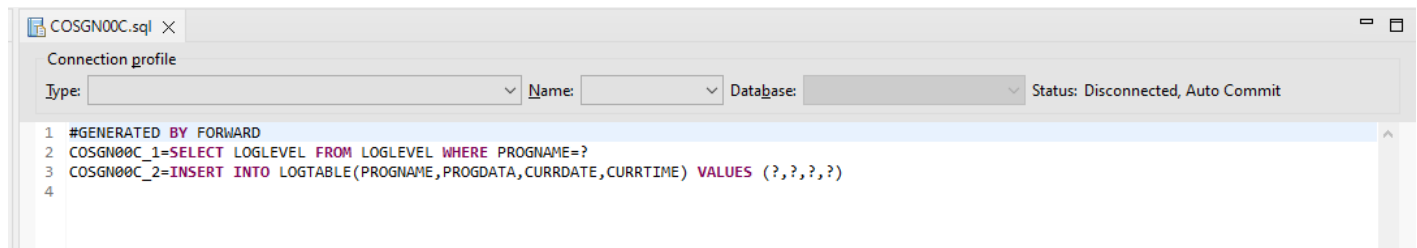
Diese Skripte werden zum Starten von Batch-Jobs (dedizierte, nicht interaktive, rechenintensive Datenverarbeitungs-Workloads) verwendet.

- SQL-Dateien

Wenn die alte Anwendung SQL-Abfragen verwendete, wurden die entsprechenden modernisierten SQL-Abfragen in speziellen Eigenschaftendateien mit dem Benennungsmuster Programm .sql gesammelt, wobei Programm der Name des Programms ist, das diese Abfragen verwendet.



Der Inhalt dieser SQL-Dateien ist eine Sammlung von Einträgen (key=query), wobei jede Abfrage einem eindeutigen Schlüssel zugeordnet ist, den das modernisierte Programm verwendet, um die angegebene Abfrage auszuführen:



Zum Beispiel führt das Programm COSGN00C die Abfrage mit dem Schlüssel „COSGN00C\_1“ (dem ersten Eintrag in der SQL-Datei) aus:

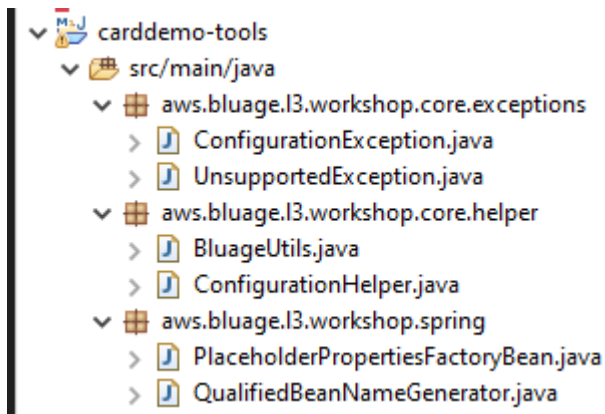
```

327- /**
328-  * Process operation getProgramLogLevel.
329-  *
330-  * *****
331-  *                GET PROGRAM LOG LEVEL
332-  * *****
333-  *
334-  * @param ctx
335-  * @param ctrl
336-  */
337- @Override
338- public void getProgramLogLevel(final Cosgn00cContext ctx, final ExecutionController ctrl) {
339-     SQLExecutorBuilder.newInstance(ctrl, ctx, ctx.getSqlca())
340-         .mapInParameter(SQLParameterBuilder.newInstance(ctx.getLogData().getLogProgramName()).type(JDBCType.CHAR))
341-         .mapOutParameter(SQLParameterBuilder.newInstance(ctx.getLogData().getLogProgramLevelReference()))
342-         .execute("COSGN00C_1");
343-     if (ctx.getSqlca().getSqlcode() == 100) {
344-         ctx.getLogData().setLogProgramLevel("N");
345-     }
346- }
347-

```

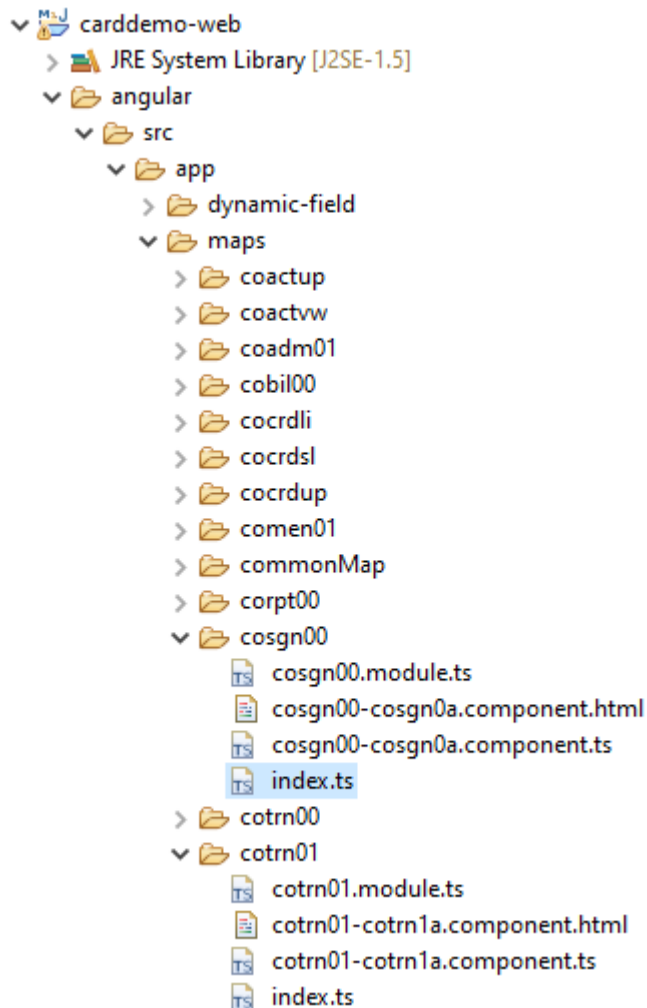
## Inhalt des Utilities-Projekts

Das Utilities-Projekt, dessen Name auf „-tools“ endet, enthält eine Reihe von technischen Hilfsprogrammen, die von allen anderen Projekten verwendet werden können.



## Inhalt des Webprojekts (e)

Das Webprojekt ist nur vorhanden, wenn ältere UI-Elemente modernisiert werden. [Die modernen UI-Elemente, die zum Aufbau des modernisierten Anwendungs-Frontends verwendet wurden, basieren auf Angular.](#) Die Beispielanwendung, die zur Darstellung der Modernisierungsartefakte verwendet wird, ist eine COBOL/CICS-Anwendung, die auf einem Mainframe läuft. Das CICS-System verwendet MAPs zur Darstellung der Benutzeroberflächenbildschirme. Die entsprechenden modernen Elemente werden für jede Map eine HTML-Datei sein, die von [Typescript-Dateien](#) begleitet wird:



Das Webprojekt kümmert sich nur um den Frontend-Aspekt der Anwendung. Das Serviceprojekt, das sich auf die Projekte Utility und Entities stützt, stellt die Backend-Services bereit. Die Verbindung zwischen dem Frontend und dem Backend erfolgt über die Webanwendung Gapwalk-Application, die Teil der Standard-Runtime-Distribution von Blu Age ist. AWS

## Programme ausführen und aufrufen

Auf älteren Systemen werden Programme als eigenständige ausführbare Dateien kompiliert, die sich selbst über einen CALL-Mechanismus aufrufen können, z. B. die COBOL CALL-Anweisung, wobei bei Bedarf Argumente übergeben werden. Die modernisierten Anwendungen bieten dieselben Funktionen, verwenden jedoch einen anderen Ansatz, da sich die Art der beteiligten Artefakte von den älteren unterscheidet.

Auf der modernisierten Seite sind Programmeinstiegspunkte spezifische Klassen, die die Programmschnittstelle implementieren. Sie sind Spring-Komponenten (`@Component`) und befinden sich in Dienstprojekten in einem Paket mit dem Namen `base package.program`



## Registrierung von Programmen

Bei jedem Start des [Tomcat-Servers](#), der modernisierte Anwendungen hostet, wird auch die Springboot-Anwendung gestartet, wodurch die Registrierung der Programme ausgelöst wird. Eine spezielle Registrierung mit dem Namen `ProgramRegistry` ist mit Programmeinträgen gefüllt, wobei jedes Programm mit seinen Bezeichnern registriert wird, ein Eintrag pro bekannter Programmkennung. Wenn ein Programm unter mehreren verschiedenen Kennungen bekannt ist, enthält die Registrierung so viele Einträge, wie es Bezeichner gibt.

Die Registrierung für ein bestimmtes Programm basiert auf der Sammlung von Bezeichnern, die von der Methode `()` zurückgegeben werden: `getProgramIdentifiers`

```

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

```

In diesem Beispiel wird das Programm einmal unter dem Namen 'CBACT04C' registriert (sehen Sie sich den Inhalt der ProgramIdentifiers-Sammlung an). Die Tomcat-Protokolle zeigen jede Programmregistrierung. Die Programmregistrierung hängt nur von den deklarierten Programmbezeichnern ab und nicht vom Programmklassennamen selbst (obwohl die Programmkennungen und Programmklassennamen normalerweise aufeinander abgestimmt sind).

Derselbe Registrierungsmechanismus gilt für Hilfsprogramme, die von den verschiedenen AWS Blu-Age-Webanwendungen bereitgestellt werden, die Teil der AWS Blu Age-Runtime-Distribution sind. Zum Beispiel stellt die Gapwalk-Utility-Pgm-Webapp die funktionalen Äquivalente der z/OS-

Systemdienstprogramme (IDCAMS, ICEGENER, SORT usw.) bereit und kann von modernisierten Programmen oder Skripten aufgerufen werden. Alle verfügbaren Hilfsprogramme, die beim Start von Tomcat registriert sind, werden in den Tomcat-Protokollen protokolliert.

## Registrierung von Skripten und Daemons

Ein ähnlicher Registrierungsprozess findet beim Start von Tomcat für Groovy-Skripte statt, die sich in der Ordnerhierarchie `/src/main/resources/scripts` befinden. Die Ordnerhierarchie für Skripts wird durchquert, und alle gefundenen Groovy-Skripten (mit Ausnahme des reservierten Skripts `special.functions.groovy`) werden in der `registriertScriptRegistry`, wobei ihr Kurzname (der Teil des Skriptdateinamens, der sich vor dem ersten Punktzeichen befindet) als Schlüssel für den Abruf verwendet wird.

### Note

- Wenn mehrere Skripten Dateinamen haben, die dazu führen, dass derselbe Registrierungsschlüssel erzeugt wird, wird nur der neueste registriert, wodurch alle zuvor aufgetretenen Registrierungen für diesen bestimmten Schlüssel überschrieben werden.
- Beachten Sie den obigen Hinweis und achten Sie bei der Verwendung von Unterordnern darauf, dass der Registrierungsmechanismus die Hierarchie flacher macht und zu unerwarteten Überschreibungen führen kann. Die Hierarchie wird bei der Registrierung nicht berücksichtigt: In der Regel führen `/scripts/a/myscript.groovy` und `/scripts/B/myscript.groovy` dazu, dass `/scripts/b/myscript.groovy` `/scripts/a/myscript.groovy` überschreiben.

Die Groovy-Skripte im Ordner `/src/main/resources/daemons` werden etwas anders behandelt. Sie sind immer noch als reguläre Skripte registriert, aber zusätzlich werden sie einmal, direkt beim Start von Tomcat, asynchron gestartet.

Nachdem die Skripte im registriert wurden `ScriptRegistry`, können sie mit einem REST-Aufruf gestartet werden, wobei die dedizierten Endpunkte verwendet werden, die die Gapwalk-Anwendung verfügbar macht. Weitere Informationen finden Sie in der entsprechenden Dokumentation.

## Programme, die Programme aufrufen

Jedes Programm kann ein anderes Programm als Unterprogramm aufrufen und ihm Parameter übergeben. Programme verwenden dazu eine Implementierung der `ExecutionController` Schnittstelle (in den meisten Fällen handelt es sich dabei um eine `ExecutionControllerImpl`

Instanz), zusammen mit einem flüssigen API-Mechanismus namens `the`, um die Argumente für den Programmaufruf `CallBuilder` zu erstellen.

Alle Programmmethoden verwenden `RuntimeContext` sowohl als auch an `ExecutionController` als Methodenargumente, sodass an immer verfügbar `ExecutionController` ist, um andere Programme aufzurufen.

Sehen Sie sich zum Beispiel das folgende Diagramm an, das zeigt, wie das Programm `CBST03A` das Programm `CBST03B` als Unterprogramm aufruft und ihm Parameter übergibt:

```

Cbstm03aProcessImpl.java x
67  /**
68   * Process operation xreffileGetNext.
69   *
70   * -----*
71   *
72   * @param ctx
73   * @param ctrl
74   */
75  @Override
76  public void xreffileGetNext(final Cbstm03aContext ctx, final ExecutionController ctrl) {
77      ctx.getWsM03bArea().setWsM03bDd("XREFFILE");
78      ctx.getWsM03bArea().setM03bRead(true);
79      DataUtils.setToZeroes(ctx.getWsM03bArea().getWsM03bRcReference());
80      DataUtils.setToBlank(ctx.getWsM03bArea().getWsM03bFldtReference());
81      ctrl.callSubProgram("CBST03B", CallBuilder.newInstance()
82          .byReference(ctx.getWsM03bArea())
83          .getArguments(), ctx);
84      if (DataUtils.compare(ctx.getWsM03bArea().getWsM03bRcReference(), "00") == 0) {
85
86          /*
87           Do nothing */
88      } else if (DataUtils.compare(ctx.getWsM03bArea().getWsM03bRcReference(), "10") == 0) {
89          ctx.getMiscVariables().setEndOfFile("Y");
90      } else {
91          if (LOGGER.isInfoEnabled()) LOGGER.info("ERROR READING XREFFILE");
92          if (LOGGER.isInfoEnabled()) LOGGER.info("{}{} ", "RETURN CODE: ", ctx.getWsM03bArea().getWsM03bRc());
93          abendProgram(ctx, ctrl);
94      }
95      ctx.getCardXrefRecord().setBytes(ctx.getWsM03bArea().getWsM03bFldtReference().getBytes());
96  }
97

```

- Das erste Argument von `ExecutionController.callSubProgram` ist ein Bezeichner des aufzurufenden Programms (d. h. einer der Bezeichner, die für die Programmregistrierung verwendet wurden — siehe Absätze oben).
- Das zweite Argument, das das Ergebnis des Builds auf der `istCallBuilder`, ist ein Array von `Record`, das den vom Aufrufer an den Aufgerufenen übergebenen Daten entspricht.
- Das dritte und letzte Argument ist die Instanz des Aufrufers `RuntimeContext`.

Alle drei Argumente sind obligatorisch und dürfen nicht Null sein, aber das zweite Argument kann ein leeres Array sein.

Der aufgerufene Benutzer kann nur dann mit übergebenen Parametern umgehen, wenn er ursprünglich dafür entworfen wurde. Für ein veraltetes COBOL-Programm bedeutet das, dass es einen LINKAGE-Abschnitt und eine USING-Klausel für die Prozedurabteilung gibt, um die LINKAGE-Elemente zu verwenden.

Sehen Sie sich zum Beispiel die entsprechende [COBOL-Quelldatei CBSTM03B.CBL](https://github.com/aws-samples/aws-mainframe-modernization-carddemo/blob/main/app/cbl/CBSTM03B.CBL) an:

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

```
98
99     LINKAGE SECTION.
100    01 LK-M03B-AREA.
101        05 LK-M03B-DD           PIC X(08).
102        05 LK-M03B-OPER        PIC X(01).
103            88 M03B-OPEN       VALUE '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
```

Das CBSTM03B-Programm verwendet also einen einzelnen Record Parameter (ein Array der Größe 1). Das ist es, was das erstellt, indem `CallBuilder` es die Verkettung der Methoden `byReference ()` und `getArguments ()` verwendet.

Die `CallBuilder` Fluent-API-Klasse verfügt über mehrere Methoden, um das Array von Argumenten aufzufüllen, die an einen Aufgerufenen übergeben werden sollen:

- `asPointer (RecordAdaptable)`: Fügt ein Argument vom Typ Zeiger als Referenz hinzu. Der Zeiger repräsentiert die Adresse einer Zieldatenstruktur.
- `byReference (RecordAdaptable)`: Fügt ein Argument als Referenz hinzu. Der Anrufer sieht die Änderungen, die der Angerufene vornimmt.
- `byReference (RecordAdaptable)`: Varargs-Variante der vorherigen Methode.

- `byValue (Object)`: Fügt ein Argument, das nach Wert in ein umgewandelt wurde, hinzu. `Record` Der Anrufer sieht die Änderungen, die der Anrufer vornimmt, nicht.
- `byValue (RecordAdaptable)`: entspricht der vorherigen Methode, aber das Argument ist direkt als verfügbar. `RecordAdaptable`
- `by ValueWith Bounds (Object, int, int)`: Fügt ein Argument hinzu, das in ein umgewandelt wird, und extrahiert den Teil des `Byte-ArraysRecord`, der durch die angegebenen Grenzen definiert ist, nach Wert.

Schließlich sammelt die Methode `getArguments` alle hinzugefügten Argumente und gibt sie als Array von zurück. `Record`

#### Note

Es liegt in der Verantwortung des Aufrufers, sicherzustellen, dass das Argumenten-Array die erforderliche Größe hat, dass die Elemente korrekt angeordnet und kompatibel sind, was das Speicherlayout angeht, mit den erwarteten Layouts für die Linkage-Elemente.

## Skripten, die Programme aufrufen

Das Aufrufen registrierter Programme aus Groovy-Skripten erfordert die Verwendung einer Klasseninstanz, die die `MainProgramRunner` Schnittstelle implementiert. Normalerweise wird das Abrufen einer solchen Instanz durch die Verwendung von `ApplicationContext` Spring erreicht:

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

Sobald eine `MainProgramRunner` Schnittstelle verfügbar ist, verwenden Sie die `RunProgram`-Methode, um ein Programm aufzurufen und den Bezeichner des Zielprogramms als Parameter zu übergeben:

```

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

```

Im vorherigen Beispiel ruft ein Job-Schritt IDCAMS (File Handling Utility Program) auf und ermöglicht so eine Zuordnung zwischen den tatsächlichen Datensatzdefinitionen und ihren logischen Bezeichnern.

Beim Umgang mit Datensätzen verwenden ältere Programme meistens logische Namen, um Datensätze zu identifizieren. Wenn das Programm von einem Skript aus aufgerufen wird, muss das Skript logische Namen den tatsächlichen physikalischen Datensätzen zuordnen. Diese Datensätze können sich im Dateisystem, in einem Blusam-Speicher befinden oder sogar durch einen Inline-Stream, die Verkettung mehrerer Datensätze oder die Generierung einer GDG definiert werden.

Verwenden Sie die `withFileConfiguration` Methode, um eine logische und physische Zuordnung von Datensätzen zu erstellen und sie dem aufgerufenen Programm zur Verfügung zu stellen.

## Schreiben Sie Ihr eigenes Programm

Es ist eine häufige Aufgabe, ein eigenes Programm zum Aufrufen von Skripten oder anderen modernisierten Programmen zu schreiben. In der Regel schreiben Sie bei Modernisierungsprojekten Ihre eigenen Programme, wenn ein ausführbares Legacy-Programm in einer Sprache geschrieben ist, die der Modernisierungsprozess nicht unterstützt, oder wenn die Quellen verloren gegangen sind (ja, das kann passieren) oder wenn es sich bei dem Programm um ein Hilfsprogramm handelt, dessen Quellen nicht verfügbar sind.

In diesem Fall müssen Sie das fehlende Programm möglicherweise selbst in Java schreiben (vorausgesetzt, Sie haben genügend Wissen über das erwartete Verhalten des Programms, das Speicherlayout der Programmargumente, falls vorhanden, usw.). Ihr Java-Programm muss den in diesem Dokument beschriebenen Programmmechaniken entsprechen, damit es von anderen Programmen und Skripten ausgeführt werden kann.

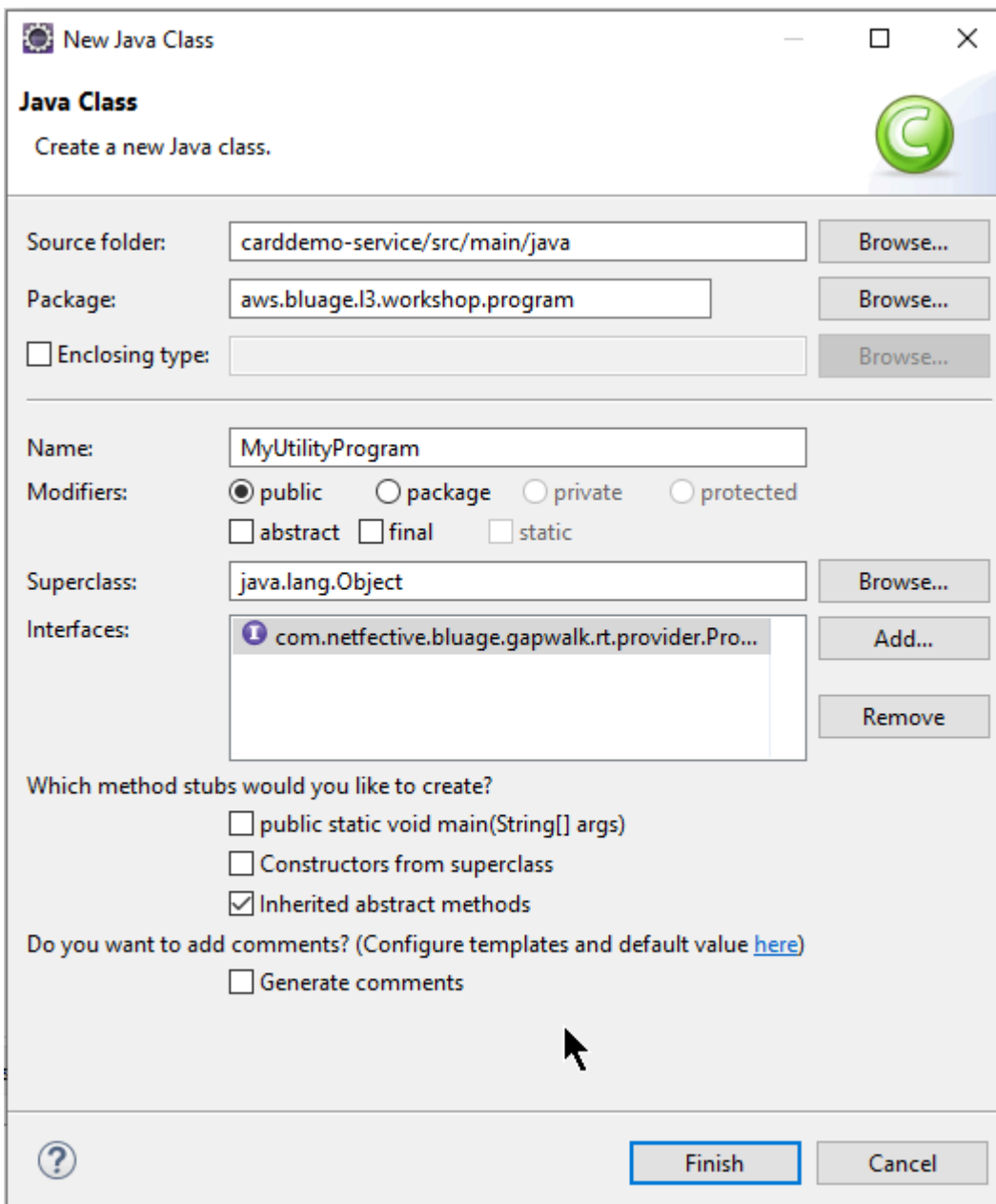
Um sicherzustellen, dass das Programm verwendet werden kann, müssen Sie zwei obligatorische Schritte ausführen:

- Schreiben Sie eine Klasse, die die Program Schnittstelle richtig implementiert, sodass sie registriert und aufgerufen werden kann.
- Stellen Sie sicher, dass Ihr Programm korrekt registriert ist, sodass es von anderen Programmen/Skripten aus sichtbar ist.

### Die Programmimplementierung schreiben

Verwenden Sie Ihre IDE, um eine neue Java-Klasse zu erstellen, die die Program Schnittstelle implementiert:





Die folgende Abbildung zeigt die Eclipse-IDE, die sich um die Erstellung aller obligatorischen Methoden kümmert, die implementiert werden müssen:

```

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

```

## Frühling-Integration

Zunächst muss die Klasse als Spring-Komponente deklariert werden. Kommentieren Sie die Klasse mit der `@Component` Anmerkung:

```

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

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

import aws.bluage.l3.workshop.SpringBootLauncher;

@Component
public class MyUtilityProgram implements Program {

```

Implementieren Sie als Nächstes die erforderlichen Methoden ordnungsgemäß. Im Zusammenhang mit diesem Beispiel haben wir das `MyUtilityProgram` zu dem Paket hinzugefügt, das bereits alle

modernisierten Programme enthält. Diese Platzierung ermöglicht es dem Programm, die bestehende Springboot-Anwendung zu verwenden, um die `ConfigurableApplicationContext` für die `getSpringApplication` Methodenimplementierung erforderlichen Informationen bereitzustellen:

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

Sie könnten einen anderen Ort für Ihr eigenes Programm wählen. Beispielsweise könnten Sie das angegebene Programm in einem anderen speziellen Serviceprojekt suchen. Stellen Sie sicher, dass das angegebene Dienstprojekt über eine eigene Springboot-Anwendung verfügt, die es ermöglicht, das abzurufen `ApplicationContext` (das sollte eine sein `ConfigurableApplicationContext`).

### Dem Programm eine Identität geben

Damit das Programm von anderen Programmen und Skripten aufgerufen werden kann, muss es mindestens eine Kennung erhalten, die nicht mit anderen registrierten Programmen innerhalb des Systems kollidieren darf. Die Wahl des Bezeichners kann darauf zurückzuführen sein, dass ein vorhandenes veraltetes Programm als Ersatz abgedeckt werden muss. In diesem Fall müssen Sie den erwarteten Bezeichner verwenden, der bei CALL-Vorgängen in den älteren Programmen angegeben ist. Die meisten Programm-IDs sind in älteren Systemen 8 Zeichen lang.

Eine Möglichkeit, dies zu tun, ist die Erstellung eines unveränderbaren Satzes von Bezeichnern im Programm. Das folgende Beispiel zeigt die Auswahl von „MYUTILPG“ als einzigen Bezeichner:

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

    public ConfigurableApplicationContext getSpringApplication() {
        I

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

### Ordnen Sie das Programm einem Kontext zu

Das Programm benötigt eine `RuntimeContext` Begleitinstanz. Für modernisierte Programme generiert AWS Blu Age automatisch den Begleitkontext unter Verwendung der Datenstrukturen, die Teil des Legacy-Programms sind.

Wenn Sie Ihr eigenes Programm schreiben, müssen Sie auch den Begleitkontext schreiben.

Unter [Kurse im Zusammenhang mit dem Programm](#) Bezugnahme auf können Sie sehen, dass ein Programm mindestens zwei Begleitklassen benötigt:

- eine Konfigurationsklasse.
- eine Kontextklasse, die die Konfiguration verwendet.

Wenn das Hilfsprogramm eine zusätzliche Datenstruktur verwendet, sollte diese ebenfalls geschrieben und vom Kontext verwendet werden.

Diese Klassen sollten sich in einem Paket befinden, das Teil einer Pakethierarchie ist, die beim Start der Anwendung gescannt wird, um sicherzustellen, dass die Kontextkomponente und die Konfiguration vom Spring-Framework verarbeitet werden.

Schreiben wir eine minimale Konfiguration und einen minimalen Kontext in das *base package.myutilityprogram.business.context* Paket, das gerade im Entitätsprojekt erstellt wurde:

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

Hier ist der Inhalt der Konfiguration. Es verwendet einen Konfigurationsaufbau, der anderen — modernisierten — Programmen in der Nähe ähnelt. Sie müssen dies wahrscheinlich an Ihre spezifischen Bedürfnisse anpassen.

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

#### Hinweise:

- Die allgemeine Benennungskonvention lautet ProgramNameKonfiguration.
- Es muss die Annotationen `@org.springframework.context.annotation.Configuration` und `@Lazy` verwenden.
- Der Bean-Name folgt normalerweise der Konvention, dies ist jedoch nicht zwingend erforderlich. `ProgramNameContextConfiguration` Achten Sie darauf, dass es im gesamten Projekt nicht zu Kollisionen mit den Bean-Namen kommt.
- Die einzige Methode, die implementiert werden soll, muss ein `Configuration` Objekt zurückgeben. Verwenden Sie die `ConfigurationBuilder` Fluent-API, um eine zu erstellen.

Und der zugehörige Kontext:

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

## Hinweise

- Die Kontextklasse sollte eine bestehende Context Schnittstellenimplementierung erweitern (entweder `RuntimeContext` oder `JicsRuntimeContext`, die um JICS-spezifische Elemente erweitert wurde `RuntimeContext`).
- Die allgemeine Benennungskonvention lautet Kontext ProgramName.
- Sie müssen es als Prototyp-Komponente deklarieren und die `@Lazy` -Annotation verwenden.
- Der Konstruktor bezieht sich auf die zugehörige Konfiguration und verwendet die `@Qualifier` -Annotation, um auf die richtige Konfigurationsklasse abzielen.
- Wenn das Hilfsprogramm einige zusätzliche Datenstrukturen verwendet, sollten diese wie folgt lauten:
  - geschrieben und dem `base package.business.model` Paket hinzugefügt.
  - im Kontext referenziert. Schauen Sie sich andere existierende Kontextklassen an, um zu erfahren, wie Sie auf Datenstrukturklassen verweisen und die Kontextmethoden (constructor/cleanup/reset) nach Bedarf anpassen können.

Nun, da ein spezieller Kontext verfügbar ist, lassen Sie das neue Programm ihn verwenden:

```

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

```

## Hinweise:

- Die getContext-Methode muss strikt wie gezeigt implementiert werden, wobei eine Delegation an die getOrCreate Methode der ProgramContextStore Klasse und das auto verdrahtete Spring BeanFactory verwendet wird. Ein einziger Programmbezeichner wird verwendet, um den Programmkontext im zu speichernProgramContextStore; auf diesen Bezeichner wird als „Hauptbezeichner des Programms“ verwiesen.
- Die zugehörigen Klassen für Konfiguration und Kontext müssen mithilfe der Spring-Annotation referenziert werden. @Import

## Implementierung der Geschäftslogik

Wenn das Programmskelett vollständig ist, implementieren Sie die Geschäftslogik für das neue Hilfsprogramm.

Tun Sie dies in der `run` Methode des Programms. Diese Methode wird immer dann ausgeführt, wenn das Programm aufgerufen wird, entweder von einem anderen Programm oder von einem Skript.

Viel Spaß beim Programmieren!

## Bearbeitung der Programm-Registrierung

Stellen Sie abschließend sicher, dass das neue Programm ordnungsgemäß in der registriert ist `ProgramRegistry`. Wenn Sie das neue Programm zu dem Paket hinzugefügt haben, das bereits andere Programme enthält, müssen Sie nichts weiter tun. Das neue Programm wird beim Start der Anwendung übernommen und bei allen Nachbarprogrammen registriert.

Wenn Sie einen anderen Speicherort für das Programm ausgewählt haben, müssen Sie sicherstellen, dass das Programm beim Start von Tomcat ordnungsgemäß registriert ist. Einige Anregungen dazu finden Sie in der Initialisierungsmethode der generierten `SpringbootLauncher` Klassen in den Dienstprojekten (siehe [Inhalt des Serviceprojekts](#)).

Überprüfen Sie die Tomcat-Startprotokolle. Jede Programmregistrierung wird protokolliert. Wenn Ihr Programm erfolgreich registriert wurde, finden Sie den passenden Logeintrag.

Wenn Sie sicher sind, dass Ihr Programm ordnungsgemäß registriert ist, können Sie mit der Iteration der Geschäftslogik beginnen.

## Vollständig qualifizierte Namenszuordnungen

Dieser Abschnitt enthält Listen der vollqualifizierten Namenszuordnungen von AWS Blu Age und Drittanbietern zur Verwendung in Ihren modernisierten Anwendungen.

### AWS Vollqualifizierte Namenszuordnungen von Blu Age

Kurzname	Vollständig qualifizierter Name
<code>CallBuilder</code>	<code>com.netfective.bluage.gapwalk.runtime.statements.CallBuilder</code>
<code>Configuration</code>	<code>com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration</code>



Kurzname	Vollständig qualifizierter Name
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

Kurzname	Vollständig qualifizierter Name
SimpleStateMachineController	com.netfective.bluage.gapwalk.rtm.statemachine.SimpleStateMachineController
StateMachineController	com.netfective.bluage.gapwalk.rtm.statemachine.StateMachineController
StateMachineRunner	com.netfective.bluage.gapwalk.rtm.statemachine.StateMachineRunner

### Vollqualifizierte Namenszuordnungen von Drittanbietern

Kurzname	Vollständig qualifizierter Name
@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

## Datenvereinfacher

Auf Mainframe- und Midrange-Systemen (im folgenden Thema als „Legacy-Systeme“ bezeichnet) ermöglichen häufig verwendete Programmiersprachen wie COBOL, PL/I oder RPG einen Low-Level-Zugriff auf Speicher. Dieser Zugriff konzentriert sich auf das Speicherlayout, auf das über systemeigene Typen wie zonenweise, gepackt oder alphanumerisch zugegriffen wird, möglicherweise auch aggregiert über Gruppen oder Arrays.

In einem bestimmten Programm gibt es eine Kombination von Zugriffen auf ein bestimmtes Speicherelement, sowohl über typisierte Felder als auch über direkten Zugriff auf Byte (Rohspeicher). COBOL-Programme übergeben beispielsweise Argumente als zusammenhängende Bytesätze (LINKAGE) an Aufrufer oder lesen und schreiben Daten aus Dateien auf die gleiche Weise (Datensätze), während sie solche Speicherbereiche mit typisierten Feldern interpretieren, die in Copybooks organisiert sind.

Solche Kombinationen aus rohem und strukturiertem Speicherzugriff, der Abhängigkeit von einem präzisen Speicherlayout auf Byteebene und veraltete Typen wie zoned oder packed sind Funktionen, die in der Java-Programmierung weder nativ noch einfach verfügbar sind.

Als Teil der AWS Blu Age-Lösung zur Modernisierung älterer Programme für Java stellt die Data Simplifier-Bibliothek solche Konstrukte für modernisierte Java-Programme bereit und stellt sie auf eine Weise zur Verfügung, die Java-Entwicklern so vertraut wie möglich ist (Getter/Setter, Byte-Arrays, klassenbasiert). Sie ist eine Kernabhängigkeit des modernisierten Java-Codes, der aus solchen Programmen generiert wird.

Der Einfachheit halber basieren die meisten der folgenden Erklärungen auf COBOL-Konstrukten. Sie können jedoch dieselbe API sowohl für die Modernisierung des PL1- als auch für das RPG-Datenlayout verwenden, da die meisten Konzepte ähnlich sind.

### Themen

- [Wichtigste Klassen](#)
- [Datenbindung und Zugriff](#)
- [Häufig gestellte Fragen zu den besprochenen Java-Typen](#)

## Wichtigste Klassen

Zur besseren Lesbarkeit verwendet dieses Dokument die Java-Kurznamen der AWS Blu Age-API-Schnittstellen und -Klassen. Weitere Informationen finden Sie unter [Häufig gestellte Fragen zu den besprochenen Java-Typen](#).

### Darstellung des Speichers auf niedriger Ebene

Auf der untersten Ebene wird der Speicher (ein zusammenhängender Bytebereich, auf den schnell und zufällig zugegriffen werden kann) durch die `Record` Schnittstelle repräsentiert. Diese Schnittstelle ist im Wesentlichen eine Abstraktion eines Byte-Arrays fester Größe. Als solches bietet es Setter und Getter, die auf die zugrunde liegenden Bytes zugreifen oder diese ändern können.

### Strukturierte Datendarstellung

Zur Darstellung strukturierter Daten wie „01 Datenelemente“ oder „01 Copybooks“, wie sie in COBOL DATA DIVISION zu finden sind, werden Unterklassen der `RecordEntity` Klasse verwendet. Diese werden normalerweise nicht von Hand geschrieben, sondern mit den Modernisierungstools von AWS Blu Age aus den entsprechenden Legacy-Konstrukten generiert. Es ist immer noch nützlich, etwas über ihre Hauptstruktur und API zu wissen, damit Sie verstehen können, wie der Code in einem modernisierten Programm sie verwendet. Im Fall von COBOL handelt es sich bei diesem Code um Java, das aus ihrer PROCEDURE DIVISION generiert wurde.

Der generierte Code stellt jedes „01-Datenelement“ mit einer `RecordEntity` Unterklasse dar. Jedes Elementarfeld oder Aggregat, aus dem es besteht, wird als privates Java-Feld dargestellt, das als Baum organisiert ist (jedes Element hat ein übergeordnetes Feld, mit Ausnahme des Stammelements).

Zur Veranschaulichung finden Sie hier ein Beispiel für ein COBOL-Datenelement, gefolgt vom entsprechenden von AWS Blu Age generierten Code, der es modernisiert:

```
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);
}
}
```

## Elementare Felder

Felder der Klasse `Elementary` (oder `Filler`, wenn sie unbenannt sind) stellen ein „Blatt“ der alten Datenstruktur dar. Sie sind einem zusammenhängenden Bereich zugrundeliegender Bytes („Bereich“) zugeordnet und haben üblicherweise einen Typ (möglicherweise parametrisiert), der angibt, wie diese Byte interpretiert und verändert werden sollen (indem jeweils ein Wert von/zu einem Byte-Array „dekodiert“ und „kodiert“ wird).

Alle elementaren Typen sind Unterklassen von `RangeType`. Übliche Typen sind:

COBOL-Typ	Typ des Datenvereinfachers
PIC X(n)	AlphanumericType
PIC 9(n)	ZonedType
PIC 9(n) COMP-3	PackedType
PIC 9(n) COMP-5	BinaryType

## Felder aggregieren

Aggregatfelder organisieren das Speicherlayout ihres Inhalts (andere Aggregate oder elementare Felder). Sie haben selbst keinen elementaren Typ.

GroupFelder stehen für zusammenhängende Felder im Speicher. Jedes der darin enthaltenen Felder ist in derselben Reihenfolge im Speicher angeordnet, wobei das erste Feld einen Offset 0 gegenüber der Gruppenfeldposition im Speicher hat, das zweite Feld einen Offset  $0 + (\text{size in bytes of first field})$  usw. Sie werden verwendet, um Sequenzen von COBOL-Feldern unter demselben enthaltenden Feld darzustellen.

UnionFelder stehen für mehrere Felder, die auf denselben Speicher zugreifen. Jedes der darin enthaltenen Felder ist in 0 Bezug auf die Position des Union-Felds im Speicher versetzt angeordnet. Sie werden beispielsweise verwendet, um das COBOL-Konstrukt „REDEFINES“ darzustellen (die ersten Union-Kinder sind das neu definierte Datenelement, die zweiten untergeordneten Elemente sind die erste Neudefinition usw.).

Array-Felder (Unterklassen von `Repetition`) stellen die Wiederholung des Layouts ihres untergeordneten Felds im Speicher dar (sei es ein Aggregat selbst oder ein elementares Element). Sie legen eine bestimmte Anzahl solcher untergeordneter Layouts im Speicher an, wobei jedes Layout einen Offset hat.  $\text{index} * (\text{size in bytes of child})$  Sie werden verwendet, um COBOL-"OCCURS"-Konstrukte darzustellen.

## Primitive

In einigen Modernisierungsfällen können „Primitive“ auch verwendet werden, um unabhängige, „Stamm“-Datenelemente darzustellen. Diese sind in ihrer Verwendung sehr ähnlich, stammen `RecordEntity` aber nicht daraus und basieren auch nicht auf generiertem Code.

Stattdessen werden sie direkt von der AWS Blu Age-Laufzeit als Unterklassen der `Primitive` Schnittstelle bereitgestellt. Beispiele für solche bereitgestellten Klassen sind `Alphanumeric` oder `ZonedDecimal`.

## Datenbindung und Zugriff

Die Verknüpfung zwischen strukturierten Daten und zugrunde liegenden Daten kann auf verschiedene Arten erfolgen.

Eine wichtige Schnittstelle für diesen Zweck ist `RecordAdaptable`, die dazu dient, eine `Record` „beschreibbare Ansicht“ auf die `RecordAdaptable` zugrunde liegenden Daten zu erhalten. Wie wir weiter unten sehen werden, werden mehrere Klassen implementiert `RecordAdaptable`. Umgekehrt erwarten AWS Blu-Age-APIs und Code, der Speicher auf niedriger Ebene manipuliert (wie Programmargumente, Datei-I/O-Datensätze, CICS-Kommunikationsbereich, zugewiesener Speicher...), häufig ein `RecordAdaptable` als Zugriffsnummer für diesen Speicher.

Im Fall der COBOL-Modernisierung sind die meisten Datenelemente mit Speicher verknüpft, der während der Laufzeit der entsprechenden Programmausführung repariert wird. Zu diesem Zweck werden `RecordEntity` Unterklassen einmal in einem generierten übergeordneten Objekt (dem Programmkontext) instanziiert und kümmern sich dann, basierend auf der `BytegrößeRecord`, um die Instanziierung der ihnen zugrunde liegenden Klassen. `RecordEntity`

In anderen COBOL-Fällen, z. B. bei der Verknüpfung von LINKAGE-Elementen mit Programmargumenten oder bei der Modernisierung des SET ADDRESS OF-Konstrukts, muss eine Instanz einer bereitgestellten Instanz zugeordnet werden. `RecordEntity RecordAdaptable` Zu diesem Zweck gibt es zwei Mechanismen:

- wenn die `RecordEntity` Instanz bereits existiert, kann die `RecordEntity.bind(RecordAdaptable)` Methode (geerbt von `Bindable`) verwendet werden, damit diese Instanz darauf „verweist“ `RecordAdaptable`. Jeder Getter oder Setter, der auf der aufgerufen wird, `RecordEntity` wird dann von den zugrundeliegenden `RecordAdaptable` Bytes unterstützt (Byte lesen oder schreiben).
- wenn der instanziiert werden soll, `RecordEntity` ist ein generierter Konstruktor verfügbar, der a akzeptiert. `RecordAdaptable`

Umgekehrt kann auf die `Record` aktuell gebundenen strukturierten Daten zugegriffen werden. Dazu `RecordEntity` implementiert `RecordAdaptable`, `getRecord()` kann also auf jeder solchen Instanz aufgerufen werden.



Schließlich benötigen viele COBOL- oder CICS-Verben Zugriff auf ein einzelnes Feld zum Lesen oder Schreiben. Die `RangeReference` Klasse wird verwendet, um einen solchen Zugriff darzustellen. Ihre Instanzen können aus `RecordEntity` generierten `getXXXReference()` Methoden (d. h. dem Feld, auf das zugegriffen XXX wird) abgerufen und an Laufzeitmethoden übergeben werden. `RangeReference` wird normalerweise für den Zugriff auf das gesamte `RecordEntity` Objekt oder verwendet `Group`, während seine Unterklasse Zugriffe auf `Elementary` Felder `ElementaryRangeReference` darstellt.

Beachten Sie, dass die meisten obigen Beobachtungen auf `Primitive` Unterklassen zutreffen, da sie darauf abzielen, ein ähnliches Verhalten zu implementieren, wie `RecordEntity` es von der AWS Blu-Age-Laufzeit (anstelle von generiertem Code) bereitgestellt würde. Zu diesem Zweck werden alle Unterklassen von `Primitive` Implementierungen `ElementaryRangeReference` und `Bindable` Interfaces so implementiert `RecordAdaptable`, dass sie anstelle von `RecordEntity` Unterklassen und elementaren Feldern verwendet werden können.

## Häufig gestellte Fragen zu den besprochenen Java-Typen

Die folgende Tabelle zeigt die vollqualifizierten Namen der in diesem Abschnitt behandelten Java-Typen.

Kurzer Name	Vollständig qualifizierter Name
<code>Alphanumeric</code>	<code>com.netfactive.bluage.gapwalk.datasimplifier.elementary.Alphanumeric</code>
<code>AlphanumericType</code>	<code>com.netfactive.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType</code>
<code>BinaryType</code>	<code>com.netfactive.bluage.gapwalk.datasimplifier.metadata.type.BinaryType</code>
<code>Bindable</code>	<code>com.netfactive.bluage.gapwalk.datasimplifier.data.Bindable</code>

Kurzer Name	Vollständig qualifizierter Name
Elementary	<code>com.netfactive.bluage.gapwalk.datasimplifier.data.structure.Elementary</code>
ElementaryRangeReference	<code>com.netfactive.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference</code>
Filler	<code>com.netfactive.bluage.gapwalk.datasimplifier.data.structure.Filler</code>
Group	<code>com.netfactive.bluage.gapwalk.datasimplifier.data.structure.Group</code>
PackedType	<code>com.netfactive.bluage.gapwalk.datasimplifier.metadata.type.PackedType</code>
Primitive	<code>com.netfactive.bluage.gapwalk.datasimplifier.elementary.Primitive</code>
RangeReference	<code>com.netfactive.bluage.gapwalk.datasimplifier.entity.RangeReference</code>
RangeType	<code>com.netfactive.bluage.gapwalk.datasimplifier.metadata.type.RangeType</code>
Record	<code>com.netfactive.bluage.gapwalk.datasimplifier.data.Record</code>

Kurzer Name	Vollständig qualifizierter Name
RecordAdaptable	<code>com.netfective.bluage.gapwalk.datasimplifier.data.RecordAdaptable</code>
RecordEntity	<code>com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity</code>
Repetition	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Repetition</code>
Union	<code>com.netfective.bluage.gapwalk.datasimplifier.data.structure.Union</code>
ZonedDecimal	<code>com.netfective.bluage.gapwalk.datasimplifier.elementary.ZonedDecimal</code>
ZonedType	<code>com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType</code>

## AWS Blu Age Runtime Konfiguration und Konfigurationsdateien

Die AWS Blu Age Runtime und der Client-Code sind Webanwendungen, die das [Spring Boot-Framework](#) verwenden. Es nutzt die Funktionen von Spring zur Bereitstellung von Konfigurationen mit mehreren möglichen Speicherorten und Prioritätsregeln. Es gibt auch ähnliche Prioritätsregeln für die Bereitstellung vieler anderer Dateien, wie Groovy-Skripte, SQL usw.

Die AWS Blu Age Runtime enthält auch zusätzliche optionale Webanwendungen, die bei Bedarf aktiviert werden können.

### Themen

- [Grundlagen der Anwendungskonfiguration](#)

- [Vorrang der Anwendung](#)
- [JNDI für Datenbanken](#)
- [AWS Geheimnisse verwenden](#)
- [Andere Dateien \(Groovy, SQL usw.\)](#)
- [Zusätzliche Webanwendung](#)
- [Eigenschaften aktivieren](#)
- [Konfiguration der Sicherheit für Gapwalk-Anwendungen](#)

## Grundlagen der Anwendungskonfiguration

Die Anwendungskonfiguration erfolgt standardmäßig über spezielle YAML-Dateien, die im `config` Ordner des Anwendungsservers bereitgestellt werden. Es gibt zwei Hauptkonfigurationsdateien für YAML:

- `application-main.yaml`
- `application-profile.yaml` (wobei der *profile* Wert während der Anwendungsgenerierung eingerichtet wird).

Die erste Datei konfiguriert das `FrameworkGapwalk-application.war`, d. h. die zweite Datei enthält zusätzliche Optionen speziell für die Client-Anwendung. Dies funktioniert bei der Verwendung von Spring-Profilen: Die Gapwalk-Anwendung verwendet das `main` Profil, während die Client-Anwendung das Profil verwendet. *profile*

Das folgende Beispiel zeigt eine typische YAML-Hauptdatei.

```
#####
#### 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:
```

Das folgende Beispiel zeigt eine typische Client-YAML-Datei.

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

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

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

Hinweise zum Inhalt der YAML-Dateien finden Sie unter. [Eigenschaften aktivieren](#)

## Vorrang der Anwendung

Für diese Konfigurationsdateien gelten die Spring-Prioritätsregeln. Insbesondere:

- Die `application-main` YAML-Datei erscheint in der Gapwalk-War-Hauptdatei mit Standardwerten, und die Datei im `config` Ordner ersetzt sie.
- Das Gleiche sollte für die Konfiguration der Client-Anwendung getan werden
- Zusätzliche Parameter können beim Serverstart über die Befehlszeile übergeben werden. Sie würden die YAML-Werte überschreiben.

Weitere Informationen finden Sie in der [offiziellen Spring Boot-Dokumentation](#).

## JNDI für Datenbanken

Die Datenbankkonfiguration wird möglicherweise mit JNDI in der Datei `context.xml` in Tomcat bereitgestellt. Jede solche Konfiguration würde die YAML-Konfiguration überschreiben. Beachten Sie jedoch, dass Sie damit Ihre Anmeldeinformationen nicht in einem geheimen Manager speichern können (siehe unten).

Das folgende Beispiel zeigt Beispielkonfigurationen für JICS und BluSam Datenbanken.

```
<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`

Wäre `jdbc/jics` für die JICS-Datenbank und `jdbc/bluesam` (achten Sie auf das 'e') für die Blusam-Datenbank.

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

Die Datenbank-URL, der Benutzername und das Passwort.

## AWS Geheimnisse verwenden

Einige der Ressourcenkonfigurationen, die Anmeldeinformationen enthalten, können durch die Verwendung von AWS Geheimnissen weiter gesichert werden. Die Idee ist, kritische Daten AWS geheim zu speichern und in der YAML-Konfiguration einen Verweis auf das Geheimnis zu haben, sodass der geheime Inhalt beim Start von Tomcat spontan ausgewählt wird.

### Geheimnisse für Aurora

Die Aurora-Datenbankkonfiguration (für Jics, Blusam, Kundendatenbank usw.) verwendet das integrierte [Datenbankgeheimnis](#), das alle relevanten Felder automatisch aus der entsprechenden Datenbank füllt.

#### Note

Der dbname Schlüssel ist optional. Abhängig von Ihrer Datenbankkonfiguration wird er in das Secret aufgenommen oder nicht. Sie können ihn dort manuell hinzufügen oder indem Sie der YAML-Datei den Namen geben.

### Andere Geheimnisse

Andere Geheimnisse beziehen sich auf Ressourcen, die ein einziges Passwort haben (insbesondere passwortgeschützte Redis-Caches). In diesem Fall muss der [andere Geheimtyp](#) mit einem einzigen Schlüssel verwendet werden. password

### YAML verweist auf Geheimnisse

application-main.yaml Sie können auf den geheimen ARN für verschiedene Ressourcen verweisen. Die wichtigsten sind:

- JICS-Datenbankanmeldedaten mit `spring.aws.jics.db.secret`
- JICS TS Queues Redis-Anmeldeinformationen mit `spring.aws.client.jics.queues.ts.redis.secret`
- Blusam-Datenbankanmeldedaten mit `spring.aws.client.bluesam.db.secret`
- Blusam-Cache-Passwort mit `spring.aws.client.bluesam.redis.secret`
- Blusam sperrt das Cache-Passwort mit `spring.aws.client.bluesam.locks.redis.secret`

Das folgende Beispiel zeigt, wie diese Geheimnisse in einer YAML-Datei deklariert werden.

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

Datenbankname: Bluesam

In diesem Beispiel ist der Name der Datenbank nicht im Secret enthalten und wird stattdessen hier angegeben.

Der Client `application-profile.yaml` kann auf den geheimen ARN für die Client-Datenbank verweisen. Dies erfordert eine zusätzliche Eigenschaft zum Auflisten der Datenquellen, wie im folgenden Beispiel dargestellt:

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



```
secret: arn:aws:secretsmanager:XXXX
```

Namen: primär, Host

Ein Beispiel mit zwei Client-Datenquellen namens `primary` und `host`, jede mit ihrer Datenbank und ihren Anmeldeinformationen.

Datenbankname: `mydb`

In diesem Beispiel ist der Name der „Host“-Datenbank nicht im Secret enthalten und wird stattdessen hier angegeben, während er für die „primäre“ Datenbank im Secret enthalten ist.

Sie können auch so konfigurieren `application-main.yaml`, dass der geheime OAuth2-Client abgerufen wird, AWS Secrets Manager indem Sie den Anbieter und den ARN angeben. Der Standardwert für die Provider-Eigenschaft ist Amazon Cognito. Im Folgenden finden Sie eine Beispielkonfiguration für den OAuth2-Anbieter Keycloak:

```
spring:
  aws:
    client:
      provider: keycloak
      keycloak:
        secret: arn:aws:secretsmanager:XXXX
```

In diesem Beispiel wird das Client-Geheimnis für den OAuth2-Anbieter Keycloak aus dem angegebenen ARN in abgerufen. AWS Secrets Manager Diese Konfiguration unterstützt mehrere Anbieter, indem der Anbieternamen und der entsprechende geheime ARN dynamisch aufgelöst werden.

`application-utility-pgm.yml` Sie können auf den geheimen ARN für verschiedene Ressourcen verweisen.

- `spring.aws.client.datasources.primary`
  - `secret` Geheimer ARN für die Anwendungsdatenbank.

Typ: Zeichenfolge

- `type`

Vollqualifizierter Name der zu verwendenden Verbindungspool-Implementierung.

Typ: Zeichenfolge

Standard: `com.zaxxer.hikari.HikariDataSource`

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

Liste der Datenquellennamen.

Typ: Zeichenfolge

- `dsname`
  - `dbname`

Name des Hosts.

Typ: Zeichenfolge

- `secret`

Geheimer ARN der Host-Datenbank.

Typ: Zeichenfolge

- `type`

Vollqualifizierter Name der zu verwendenden Verbindungspool-Implementierung.

Typ: Zeichenfolge

Standard: `com.zaxxer.hikari.HikariDataSource`

Für ein Geheimnis aus mehreren Datenquellen:

```
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
```

### Keine von XA unterstützten geheimen Schlüssel

- Engine (postgres/oracle/db2/mssql)
- port
- dbname
- Aktuelles Schema
- username
- password
- URL
- SSL-Verbindung
- SSL-Standort TrustStore
- TrustStoressl-Passwort

Denn postgres nur der sslMode geheime Schlüssel value (disable/allow/prefer/require/verify-ca/verify-full) ermöglicht zusätzlich zur spring.aws.rds.ssl.cert-path YAML-Eigenschaft die Verbindung mit SSL.

## XA unterstützte geheime Schlüssel

Wenn die Client-Datenbank XA verwendet, werden die Sub-XA-Eigenschaften durch geheime Werte unterstützt.

- Host
- port
- dbname
- Aktuelles Schema
- username
- password
- URL
- SSL-Verbindung (wahr/falsch)
- TrustStoreSSL-Standort
- TrustStoressl-Passwort

Für andere XA-Eigenschaften (zum Beispiel maxPoolSize oderdriverType) spring.jta.atomikos.datasource.XXXX.unique-resource-name muss der reguläre YAML-Schlüssel jedoch weiterhin angegeben werden.

Der geheime Wert hat Vorrang vor den YAML-Eigenschaften.

## Andere Dateien (Groovy, SQL usw.)

Die anderen vom Kundenprojekt verwendeten Dateien verwenden ähnliche Prioritätsregeln wie die für die Spring-Konfiguration. Beispiele:

- Groovy-Skripte sind .groovy Dateien im scripts Ordner oder in Unterordnern.
- SQL-Skripten sind .sql Dateien im sql Ordner oder in Unterordnern.
- Daemon-Skripten sind .groovy Dateien im daemons Ordner oder in Unterordnern.
- Abfragen Bei der Datenbankzuordnungsdatei handelt es sich um Dateien mit dem Namen queries-database.mapping Dateien in den sql Unterordnern des Ordners.

- Jasper-Vorlagen sind `.jrxml` Dateien im `templates` Ordner oder in Unterordnern.
- Datensatzkataloge sind `.json` Dateien im Ordner `catalog`
- Lnk-Dateien sind `.json` Dateien im `lnk` Ordner.

All diese Speicherorte können mithilfe einer Systemeigenschaft oder einer Client-YAML-Eigenschaft außer Kraft gesetzt werden.

- Für Groovy-Skripte: `configuration.scripts`
- Für SQL-Skripte: `configuration.sql`
- Für Daemon-Skripte: `configuration.daemons`
- Für Abfragen: Datenbank-Mapping-Datei: `configuration.databaseMapping`
- Für Jasper-Vorlagen: `configuration.templates`
- Für Dataset-Kataloge: `configuration.catalog`
- Für Lnk-Dateien: `configuration.lnk`

Wenn die Eigenschaft nicht gefunden wird, werden die Dateien vom oben genannten Standardspeicherort übernommen. Die Suche erfolgt zunächst mit dem Tomcat-Arbeitsverzeichnis als Stammverzeichnis und zuletzt in der WAR-Datei der Anwendung.

## Zusätzliche Webanwendung

Die AWS Blu Age Runtime enthält zusätzliche Webanwendungen in ihrem `webapps-extra` Ordner. Diese Anwendungen werden standardmäßig nicht vom Tomcat-Server bereitgestellt.

Die Anmeldung für diese Webanwendungen hängt vom Modernisierungsprojekt ab und erfolgt durch Verschieben der gewünschten WAR-Datei aus dem Ordner in den `webapps-extra` Ordner. Danach wird der Krieg beim nächsten Start vom Tomcat-Server bedient.

Einige projektspezifische Zusatzkonfigurationen können auch in einer YAML-Konfigurationsdatei für jeden weiteren Krieg hinzugefügt werden, wie in der `application-main.yml` Datei beschrieben und oben erklärt. Die zusätzlichen Kriege sind:

- `gapwalk-utility-pgm.war`: enthält Unterstützung für ZOS-Hilfsprogramme und wird `application-utility-pgm.yml` als Konfiguration verwendet.
- `gapwalk-cl-command.war`: enthält Unterstützung für AS/400-Hilfsprogramme und wird `application-cl-command.yml` als Konfiguration verwendet.

- `gapwalk-hierarchical-support.war`: enthält IMS/MFS-Transaktionsunterstützung und wird als Konfiguration verwendet `application-jhdb.yaml`

## Eigenschaften aktivieren

In Spring Boot-Anwendungen `application-main.yml` handelt es sich um die Konfigurationsdatei, in der wir verschiedene Arten von Eigenschaften wie den Listening-Port, die Datenbankkonnektivität und vieles mehr definieren.

### Themen

- [YML-Notation](#)
- [Schnellstart//Anwendungsfälle](#)
- [Verfügbare Eigenschaften für die Hauptanwendung](#)
- [Verfügbare Eigenschaften für optionale Webanwendungen](#)

## YML-Notation

In der folgenden Dokumentation `parent.child1.child2=true` wird eine Eigenschaft wie folgt im YAML-Format geschrieben.

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

## Schnellstart//Anwendungsfälle

Die folgenden Anwendungsfälle zeigen Beispiele für die entsprechenden Schlüssel und Werte.

- Standard `application-main.yml`

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

#####
##### Logging configuration #####
#####
```

```

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

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

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

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

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

#####
##### Level 2 cache #####

```

```

#####
#       cache:
#       use_second_level_cache: true
#       use_query_cache: true
#       region:
#       factory_class: org.hibernate.cache.ehcache.EhCacheRegionFactory
#   javax:
#       persistence:
#       sharedCache:
#       mode: ENABLE_SELECTIVE
#####
##### Redis settings #####
#####
    session:
        store-type: none #redis

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

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

```



```
#####  
##### Embedded Bluesam configuration #####  
#####  
bluesam :  
  remote : false  
  cache : ehcache  
  persistence : pgsql #pgsql, mssql, xodus...  
  ehcache:  
    resource-pool:  
      size: 4GB  
  write-behind:  
    enabled: true  
  pgsql :  
    dataSource : bluesamDs  
  
#####  
##### Jics settings #####  
#####  
rabbitmq.host: localhost  
jics:  
  cache: false #redis  
  resource-definitions.store-type: jpa # default value: jpa, other possible value:  
  redis  
  redis.hostname: 127.0.0.1 # Redis server host.  
  redis.password: redis # Login password of the redis server.  
  redis.port: 6379 # Redis server port.  
  redis.username: # Redis username  
  redis.mode: standalone # Redis mode. Possible values: standalone, cluster  
jics.disableSyncpoint : false  
#jics.initList:  
#jics.parameters.datform: DDMMYY  
#jics.parameters.applid: VELOCITY  
#jics.parameters.sysid: CICS  
#jics.parameters.eibtrmid: TERM  
#jics.parameters.userid: MYUSERID  
#jics.parameters.username: MYUSERNAME  
#jics.parameters.opid: XXX  
#jics.parameters.cwa.length: 0  
#jics.parameters.netname: MYNETNAME  
#jics.parameters.jobname: MJOBNAME  
#jics.parameters.sysname: SYSNAME  
  
#####
```

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

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

#####
##### DateHelper configuration #####
#####
#forcedDate: "2013-08-26T12:59:58+01:57"

#####
##### Sort configuration #####
#####
#externalSort.threshold: 256MB

#####
##### Server timeout (10 min) #####
#####
spring.mvc.async.request-timeout: 600000

#####
##### DATABASE STATISTICS #####
#####
databaseStatistics : false

#####
##### CALLS GRAPH #####
#####
callGraph : false

#####
##### SQL SHIFT CODE POINT #####
#####
# Code point 384 match unicode character \u0180
sqlCodePointShift : 384

#####
```

```
##### LOCK TIMEOUT RECORD #####
#####
# Blu4IV record lock timeout
lockTimeout : 100

#####
##### REPORTS OUTPUT PATH #####
#####
reportOutputPath: reports

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

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

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

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

#####
```

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

- Verwenden Sie Dateien mit variabler Länge mit LISTCAT-Befehlen

```
[**/*. *]
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)
```

- Geben Sie im LOAD/UNLOAD-Hilfsprogramm einen Null-Byte-Indikatorwert an

```
# 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
```

## Verfügbare Eigenschaften für die Hauptanwendung

Diese Tabelle bietet einen umfassenden Überblick über die Schlüssel-/Werteparameter.

Schlüssel	Typ	Standardwert	Beschreibung
<code>logging.config</code>	Pfad	Klassenpfad: <code>logback-main.xml</code>	Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar.
<code>spring.jta.enabled</code>	boolesch	<code>false</code>	Standardschlüssel. Wenn der Modus zur Unterstützung der Datenquelle nicht <code>static-xa</code> ist, muss die automatische Konfiguration von Spring-JTA-Transaktionen deaktiviert werden.
<code>datasource.jicsDs + -driver-class-name + -url + -username + -password + -type</code>	Standard-Spring-Datenquelle mit Unterschlüsseln		Enthält die Verbindungsinformationen für die Jics-Datenbank. Alternativ wird die Verwendung von AWS-Geheimnissen dringend empfohlen, wie in <a href="#">xref: erklärt. /configuration/configuration</a>

Schlüssel	Typ	Standardwert	Beschreibung
			n.adoc [Konfiguration].
datasource.bluesamDs + -driver-class-name + -url + -username + -password + -type	Standard-Spring-Datenquelle mit Unterschlüsseln		Enthält die Verbindungsinformationen für die Blusam-Datenbank. Alternativ wird die Verwendung von AWS-Geheimnissen dringend empfohlen, wie in xref: erklärt. /configuration/configuration.n.adoc [Konfiguration].
bluesam.disabled	boolesch	false	Ob Blusam vollständig deaktiviert werden soll.
bluesam.cache	Zeichenfolge		Wenn nicht gesetzt, wird der Blusam-Cache nicht verwendet. Mögliche Werte (Cache-Implementierungen) sind cache und redis.
forcedDate	Zeichenfolge		Erzwingt das Datum auf das angegebene Datum, falls es eines gibt.

Schlüssel	Typ	Standardwert	Beschreibung
frozenDate	boolesch	true	Gibt an, ob das Datum eingefroren werden soll. Gilt nur, wenn forcedDate es auch gesetzt ist.
externalSort.threshold	Datengröße (Beispiel: 12 MB)		Der Schwellenwert für die Sortierung: wann zur externen Sortierung (Zusammenführung) gewechselt werden soll.
jics.parameters.datform	Zeichenfolge	MMDDYY	Das Datumsformular.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.initList</code>	Zeichenfolge		<p>Die JICS-Initialisierungsliste, durch Kommas getrennt. Falls vorhanden, definiert sie kommasetrennte Namen von Listen, die beim Start von Apache Tomcat aktiviert werden sollen, in den CICS-Listen. Beispielwert: \$UUU,DFH \$IVPL,PEZ1</p> <p>Dies wird auf Gruppen übertragen, die in diesen Listen und ihren zugrunde liegenden Ressourcendefinitionen enthalten sind, die dann für die Laufzeit sichtbar sind. Standardmäßig leer.</p>
<code>jics.parameters.applid</code>	Zeichenfolge	GESCHWINDIGKEIT	Die zur Identifizierung verwendete Anwendung in JICS (mindestens 4 Zeichen, keine maximale Länge).
<code>jics.parameters.sysid</code>	Zeichenfolge	CICS	Die Systemidentifikation (SYSID).



Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.parameters.eibtrmid</code>	Zeichenfolge	BEGRIFF	Die Terminal-ID (maximal 4 Zeichen, mindestens 1 Zeichen).
<code>jics.parameters.userid</code>	Zeichenfolge		Die Benutzer-ID (maximal 8 Zeichen, kein Minimum). Wenn kein Wert angegeben wird (standardmäßig leer), wird die HTTP-Sitzungs-ID als Benutzer-ID verwendet.
<code>jics.parameters.username</code>	Zeichenfolge	MEIN NUTZERNAME	Der Benutzername (maximal 10 Zeichen, mindestens 1).
<code>jics.parameters.netname</code>	Zeichenfolge	.NET/NAME	Der Netzwerkname (maximal 8 Zeichen, mindestens 1 Zeichen).
<code>jics.parameters.opid</code>	Zeichenfolge	XXX	Die dreistellige Bedieneridentifikation.
<code>jics.parameters.jobname</code>	Zeichenfolge	JOBNAME	Der Jobname.
<code>jics.parameters.sysname</code>	Zeichenfolge	SYSNAME	Der AS400-Systemname (sysname).

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.parameters.cwa.length</code>	Zahl	0	Die Länge des gemeinsamen Arbeitsbereichs (CWA).
<code>jics.parameters.charset</code>	Zeichenfolge	CP037	Weltweit verwendeter JICS-Zeichensatz.
<code>jics.parameters.tsqimpl</code>	Zeichenfolge	Bluesam	Implementierung von JICS Temporary Storage Queue (TSQ) (zulässige Werte sind//) bluesam memory redis
<code>jics.queues.ts.redis.hostname</code>	Zeichenfolge	127.0.0.1	Der Hostname des Jics-Cache-Redis-Servers.
<code>jics.queues.ts.redis.port</code>	Zahl	6379	Der Port des Jics-Cache-Redis-Servers.
<code>jics.queues.ts.redis.password</code>	Zeichenfolge	redis	Das Passwort für den Jics-Cache-Redis-Server.
<code>jics.queues.ts.redis.username</code>	Zeichenfolge		Der Benutzername für den Jics-Cache-Redis-Server. Die Standardeinstellung ist leer (kein Benutzername).

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.queues.redis.mode</code>	Zeichenfolge	eigenständige	Der Jics-Cache-Modus. Mögliche Werte sind <code>standalone</code> oder <code>cluster</code> . Die Standardeinstellung ist <code>standalone</code> .
<code>lockTimeout</code>	Zahl	500	Das Sperr-Timeout in Millisekunden.
<code>sqlCodePointShift</code>	Zahl		Optional. Die Verschiebung des SQL-Codepunkts. Verschiebt den Codepunkt für Steuerzeichen, auf die wir bei der Migration älterer RDBMS-Daten auf ein modernes RDBMS stoßen könnten. Sie können beispielsweise angeben, dass es dem Unicode-Zeichen entspricht. <code>384 \u0180</code>
<code>sqlIntegerOverflowAllowed</code>	boolesch	false	Gibt an, ob der SQL-Ganzzahlüberlauf zulässig ist, d. h. ob das Platzieren größerer Werte in der Hostvariablen zulässig ist.

Schlüssel	Typ	Standardwert	Beschreibung
<code>database.cursor.overflow.allowed</code>	boolesch	true	Gibt an, ob der Cursorüberlauf zugelassen werden soll. Wird auf <code>true</code> gesetzt, um den nächsten Aufruf des Cursors unabhängig von seiner Position auszuführen. Legt fest, ob überprüft werden <code>false</code> soll, ob sich der Cursor an der letzten Position befindet, bevor ein nächster Cursor aufgerufen wird. Nur aktivieren, wenn der Cursor SCROLLABLE (SENSITIVE oder INSENSITIVE) ist.
<code>reportOutputPath</code>	Zeichenfolge	<code>/reports</code>	Der Ausgabepfad des Berichts.
<code>spring.session.store-type</code>	Zeichenfolge	Keine	Der Sitzungscache für Umgebungen mit hoher Verfügbarkeit. Mögliche Werte sind <code>none</code> oder <code>redis</code> . Die Standardeinstellung ist <code>none</code> .

Schlüssel	Typ	Standardwert	Beschreibung
<code>stopExecutionWhenProgramNotFound</code>	boolesch	true	Gibt an, ob die Ausführung beendet werden soll, wenn ein Programm nicht gefunden wird. Wenn auf <code>true</code> gesetzt, wird die Ausführung unterbrochen, wenn kein Programm gefunden wird.
<code>forceHR</code>	boolesch	false	Gibt an, ob menschenlesbares SYSPRINT entweder für die Konsolen- oder Dateiausgabe verwendet werden soll.
<code>rollbackOnRTE</code>	boolesch	false	Gibt an, ob eine implizite Run-Unit-Transaktion bei Laufzeitausnahmen rückgängig gemacht werden soll.
<code>sctThreadLimit</code>	long	5	Das Thread-Limit für das Auslösen von Skripten.

Schlüssel	Typ	Standardwert	Beschreibung
<code>dataSimplifier.onInvalidNumericData</code>	Zeichenfolge	ablehnen	Wie reagiert man, wenn man ungültige numerische Daten dekodiert? Zulässige Werte sind <code>reject</code> / <code>toleratespaces</code> / <code>toleratespaceslow</code> / <code>alues</code> / <code>toleratemost</code> . Die Standardinstellung ist <code>reject</code> .
<code>filesDirectory</code>	Zeichenfolge		Das Verzeichnis für Batch-Eingabe-/Ausgabedateien.
<code>ims.messages.extendedSize</code>	boolesch	false	Gibt an, ob die erweiterte Größe für IMS-Nachrichten festgelegt werden soll.
<code>defaultKeepExistingFiles</code>	boolesch	false	Gibt an, ob der vorherige Standardwert der Datenmenge festgelegt werden soll.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.db.ddlScriptLocation</code>	Zeichenfolge		Der Speicherort des Jics-DDL-Skripts. Ermöglicht es Ihnen, das Jics-Datenbankschema mithilfe eines SQL-Skripts zu initiieren. Standardmäßig leer. Beispiel: <code>./jics/sql/jics.sql</code>
<code>jics.db.schemaTestQueryLocation</code>	Zeichenfolge		Speicherort der SQL-Datei, die eine eindeutige Abfrage enthalten soll, die die Anzahl der Objekte im JICS-Schema (falls vorhanden) zurückgibt.
<code>jics.db.dataScriptLocation</code>	Zeichenfolge		Speicherort des Skripts <code>initJics.sql</code> , das von Analyzer anhand der Analyse von CSD-Exporten vom Mainframe erstellt wurde.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.db.dataTestQueryLocation</code>	Zeichenfolge		Speicherort eines SQL-Skripts, das eine einzelne SQL-Abfrage enthält, von der erwartet wird, dass sie eine Anzahl von Objekten zurückgibt (z. B. das Zählen der Datensätze in der JICS-Programmtabelle). Wenn die Anzahl gleich 0 ist, wird die Datenbank mit dem <code>jics.db.dataScriptLocation</code> Skript geladen, andernfalls wird das Laden der Datenbank übersprungen.
<code>jics.data.dataJsonInitLocation</code>	Zeichenfolge		
<code>jics.xa.agent.timeout</code>	Zahl		
<code>query.useConcatCondition</code>	boolesch	false	Gibt an, ob die Schlüsselbedingung durch Schlüsselverkettung erstellt wird oder nicht.
<code>system.qdecfmt</code>	Zeichenfolge		



Schlüssel	Typ	Standardwert	Beschreibung
<code>disposition.checkexistence</code>	boolesch	false	Gibt an, ob die Überprüfung der Existenz einer Datei für Dataset mit DISP SHR oder OLD freigegeben werden soll.
<code>useControlMVariable</code>	boolesch	false	Gibt an, ob die Control-M-Spezifikation für die Variablenersetzung verwendet werden soll.
<code>card.encoding</code>	Zeichenfolge	CP1145	Kartenkodierung: zur Verwendung mit <code>useControlMVariable</code>
<code>mapTransformation.prefixes</code>	Zeichenfolge	<code>&amp;,@,%%</code>	Liste der Präfixe, die bei der Transformation von ControlM-Variablen verwendet werden sollen. Jedes ist durch ein Komma getrennt.
<code>checkinputfilesize</code>	boolesch	false	Gibt an, ob eine Überprüfung ausgelöst werden soll, wenn die Dateigröße ein Vielfaches der Datensatzgröße ist.

Schlüssel	Typ	Standardwert	Beschreibung
<code>stepFailWhenAbend</code>	boolesch	true	Gibt an, ob ein Ende ausgelöst werden soll, wenn die Ausführung eines Schritts fehlschlägt oder abgeschlossen wird.
<code>bluesam.fileLoading.commitInterval</code>	Zahl	100000	Das Bluesam-COMMIT-Intervall.
<code>uppercaseUserInput</code>	boolesch	true	Gibt an, ob Benutzereingaben in Großbuchstaben erfolgen müssen.
<code>jhdb.lterm</code>	Zeichenfolge		Ermöglicht es Ihnen, bei einer IMS-Emulation eine gemeinsame logische Terminal-ID zu erzwingen. Wenn nicht gesetzt, wird <code>sessionId</code> verwendet.
<code>jhdb.identificationCardData</code>	Zeichenfolge		Wird verwendet, um einige „Bediener-Identifikationsdaten“ fest in das durch den CARD-Parameter angegebene MID-Feld zu codieren. Standardmäßig leer, keine Eingabebeschränkung.

Schlüssel	Typ	Standardwert	Beschreibung
encoding	Zeichenfolge	ASCII	Die in Projekten verwendete Kodierung (nicht in Groovy-Dateien). Erwartet eine gültige Kodierung CP1047,,IBM930,... ASCII UTF-8
cl.configuration.context.encoding	Zeichenfolge	CP297	Die Kodierung von CL-Dateien. Erwartet eine gültige Kodierung CP1047IBM930,,ASCII,UTF-8. Der Standardwert ist CP297
cl.zonedMode	Zeichenfolge	EBCDIC_STRICT	Der Modus für die Kodierung oder Dekodierung von CL-Befehlen (Control Language). Zulässige Werte sind EBCDIC_STRICT /EBCDIC_MODIFIED /AS400.

Schlüssel	Typ	Standardwert	Beschreibung
<code>ims.programs</code>	Zeichenfolge		Liste der zu verwendenden IMS-Programme. Trennen Sie jeden Parameter durch ein Semikolon (;) und jede Transaktion durch ein Komma (,) . , Beispiel:PCP008,PC T008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;
<code>jhdb.configuration.context.encoding</code>	Zeichenfolge	CP297	Die JHDB-Kodierung (Java Hierarchi cal Database). Erwartet eine gültige KodierungszeichenfolgeCP1047,,,... IBM930 ASCII UTF-8
<code>jhdb.metadata.extrapath</code>	Zeichenfolge	datei:./setup/	Ein Konfigurationsparameter, der einen zusätzlichen, laufzeitspezifischen Stammordner für psbs- und dbds-Ordner angibt.

Schlüssel	Typ	Standardwert	Beschreibung
jhdb.checkpointPersistence	Zeichenfolge	Keine	<p>Der Checkpoint-Persistenzmodus. Zulässige Werte sind none//add. end. Wird verwendet add, um Checkpoints beizubehalten, wenn ein neuer erstellt und der Registrierung hinzugefügt wird. Wird verwendet, end um Checkpoints beim Herunterfahren des Servers beizubehalten. Alle anderen Werte deaktivieren die Persistenz. Beachten Sie, dass jedes Mal, wenn ein neuer Checkpoint zur Registrierung hinzugefügt wird, alle vorhandenen Checkpoints serialisiert und die Datei gelöscht wird. Es handelt sich nicht um eine Anlage an die vorhandenen Daten in der Datei. Je nach Anzahl der Checkpoints kann es also einige Auswirkungen auf die Leistung haben.</p>

Schlüssel	Typ	Standardwert	Beschreibung
jhdb.chec kpointPath	Zeichenfolge	datei:. /setup/	jhdb.chec kpointPer sistence Ist none dies nicht der Fall, können Sie mit diesem Parameter den Checkpoint t-Persistenzpfad (Speicherort der Datei checkpoin t.dat) einrichten. Alle in der Registrie rung enthaltenen Checkpoint-Daten werden serialisiert und in einer Datei (checkpoint.dat) gesichert, die sich im angegebenen Ordner befindet. Beachten Sie, dass nur Checkpoint-Daten (ScriptID, StepID, Datenbankposition und Checkpoint- Bereich) von dieser Sicherung betroffen sind.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jhdb.navigation.cacheNexts</code>	Zahl	5000	Die Cache-Dauer (in Millisekunden), die bei der hierarchischen Navigation für ein RDBMS verwendet wird.
<code>jhdb.use-db-prefix</code>	boolesch	true	Gibt an, ob ein Datenbankpräfix in der hierarchischen Navigation für ein RDBMS aktiviert werden soll.
<code>jhdb.query.limitJoinUsage</code>	boolesch	true	Gibt an, ob der Parameter Limit Join-Verwendung in RDBMS-Diagrammen verwendet werden soll.
<code>taskExecutor.corePoolSize</code>	Zahl	5	Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter , um die Größe des Core-Pools einzurichten.

Schlüssel	Typ	Standardwert	Beschreibung
<code>taskExecutor.maxPoolSize</code>	Zahl	10	Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter , um die maximale Poolgröße (maximale Anzahl parallel Threads) einzurichten.
<code>taskExecutor.queueCapacity</code>	Zahl	50	Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter , um die Warteschlangenlänge einzurichten. (= maximale Anzahl ausstehender Transaktionen, wenn sie erreicht <code>taskExecutor.maxPoolSize</code> ist)



Schlüssel	Typ	Standardwert	Beschreibung
<code>taskExecutor.allowCoreThreadTimeOut</code>	boolesch	false	Gibt an, ob ein Timeout für Kern-Threads in JCIS zulässig ist. Dies ermöglicht dynamisches Wachsen und Verkleinern auch in Kombination mit einer Warteschlange, die nicht Null ist (da die maximale Poolgröße erst zunimmt, wenn die Warteschlange voll ist).
<code>jics.runUnitLauncherPool.enable</code>	boolesch	false	Gibt an, ob der Run Unit Launcher-Pool in JICS aktiviert werden soll.
<code>jics.runUnitLauncherPool.size</code>	Zahl	20	Die Größe des Run-Unit-Launcher-Pools in JICS.
<code>jics.runUnitLauncherPool.validationInterval</code>	Zahl	1000	Das Intervall zwischen den einzelnen Ausführungen der Aufgabe, durch das die Größe des Pools angepasst wird.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.runUnitLauncherPool.parallelism</code>	Zahl	2	Die Anzahl der Threads, die verwendet werden, um die fehlenden Instanzen in der Warteschlange zu erzeugen, wenn die Anpassungsaufgabe ausgeführt wird.
<code>context.p reconstruct.enable</code>	boolesch	false	Gibt an, ob die Vorkonstruktion des Programmkontextes aktiviert werden soll.
<code>context.p reconstruct.frequencyInMillis</code>	Zahl	100	Das Intervall zwischen den einzelnen Ausführungen der Aufgabe, durch das die Größe des Pools angepasst wird.
<code>context.p reconstruct.parallelism</code>	Zahl	5	Die Anzahl der Threads, die verwendet werden, um die fehlenden Instanzen in der Warteschlange zu erzeugen, wenn die Anpassungsaufgabe ausgeführt wird.

Schlüssel	Typ	Standardwert	Beschreibung
<code>context.p reconstru ct.minIns tances</code>	Zahl	2	Die Anzahl der Instanzen, die erstellt werden, wenn ein Kontext zum ersten Mal benötigt wird.
<code>spring.aw s.applica tion.cred entials</code>	Zeichenfolge	Null	Laden Sie die AWS-Anmeldeinformationen aus der Datei mit den Anmeldeinformationsprofilen in JICS.
<code>jics.queu es.sqs.region</code>	Zeichenfolge	eu-west-1	Die AWS-Region für AWS Simple Queue Service, die in JICS verwendet wird.
<code>mq.queues .sqs.region</code>	Zeichenfolge	eu-west-3	Die AWS-Region für den AWS SQS MQ-Service.
<code>quartz.sc heduler.stand- by-if-error</code>	boolesch	false	Gibt an, ob die Auftragsausführung ausgelöst werden soll, wenn sich der Job-Scheduler im Standby-Modus befindet. Falls wahr, wird die Jobausführung nicht ausgelöst, wenn diese Option aktiviert ist.

Schlüssel	Typ	Standardwert	Beschreibung
databaseStatistics	boolesch	false	Gibt an, ob SQL-Build er Statistkinformationen sammeln und anzeigen dürfen.
dbDateFormat	Zeichenfolge	yyyy-MM-dd	Das Zieldatumsformat der Datenbank.
dbTimeFormat	Zeichenfolge	HH:MM:SS	Das DB-Zielzeitformat.
dbTimestampFormat	Zeichenfolge	yyyy-MM-DD hh:mm:ss.ssssss	Das Zeitstempelformat für das Datenbankziel.
dateTimeFormat	Zeichenfolge	ISO	Das dateTimeFormat beschreibt, wie der Datums- und Uhrzeitstempeltyp einer Datenbank in Data Simplifier-Entitäten übertragen wird. Zulässige Werte sind ISO////EUREURUSALOCAL
localDateFormat	Zeichenfolge		Liste der lokalen Datumsformate. Trennen Sie jedes Format durch \
localTimeFormat	Zeichenfolge		Liste der lokalen Zeitformate. Trennen Sie jedes Format durch \

Schlüssel	Typ	Standardwert	Beschreibung
localTime stampFormat	Zeichenfolge		Liste der lokalen Zeitstempelformate . Trennen Sie jedes Format durch \.
pgmDateFormat	Zeichenfolge	yyyy-MM-dd	Das Datums- und Uhrzeitformat.
pgmTimeFormat	Zeichenfolge	Hh.mm.ss	Das für die Ausführung von PGM-Progr ammen (Programm en) verwendete Zeitformat.
pgmTimest ampFormat	Zeichenfolge	yyyy-MM-dd-hh.mm.s s.ssssss	Das Zeitstemp elformat.
cacheMetadata	boolesch	true	Gibt an, ob Datenbank-Metadate n zwischengespeicher t werden sollen.
forceDisa bleSQLTri mStringType	boolesch	false	Gibt an, ob das Kürzen aller SQL- Zeichenkettenp arameter deaktiviert werden soll.
fetchSize	Zahl		Der FetchSize-Wert für Cursor. Wird verwendet, wenn Daten mithilfe von Chunks mithilfe von Load/Unload-Utils abgerufen werden.

Schlüssel	Typ	Standardwert	Beschreibung
<code>check-groovy-file</code>	boolesch	true	Gibt an, ob der Inhalt von Groovy-Dateien vor der Registrierung überprüft werden soll.
<code>qtemp.uuid.length</code>	Zahl	9	Die Länge der eindeutigen QTEMP-ID.
<code>qtemp.dblog</code>	boolesch	false	Ob die QTEMP-Datenbankprotokollierung aktiviert werden soll.
<code>qtemp.cleanup.threshold.hours</code>	Zahl	0	Um anzugeben, wann aktiviert <code>qtemp.dblog</code> ist. Die Lebensdauer der DB-Partition (in Stunden).
<code>sort.function</code>	Zeichenfolge		Der Name der Sortierfunktion für die blu4iv-Datenbank.
<code>invalidDataTolerance</code>	boolesch	true	Gibt an, ob ungültige Daten für den gepackten Typ toleriert werden.

Schlüssel	Typ	Standardwert	Beschreibung
<code>program.timeout</code>	Zahl	-1	Gibt eine Zeitüberschreitung in Sekunden für die Ausführung von Programmen oder Transaktionen an. Nach Ablauf dieser Zeit versucht das System, das Programm zu unterbrechen.
<code>gapwalk.line.separator</code>	Zeichenfolge	Null	Gibt den Typ des Zeilentrennzeichens in Gapwalk an. Die zulässigen Werte sind WIN (CRLF)/UNIX (LF)/LINUX (LF). Andere Werte werden ignoriert und die Systemeigenschaft <code>line.separator</code> wird verwendet.

Schlüssel	Typ	Standardwert	Beschreibung
<code>enableActivePgmIdCache</code>	boolesch	false	Gibt an, ob der aktive lokale Programm-ID-Cache aktiviert werden soll. Verwenden Sie diese Funktion mit Vorsicht, da JICS-Ressourcen von Programmen und Benutzern gemeinsam genutzt werden können. Diese Ressourcen können von jedem Administrator extern geändert werden, und der installierte lokale Cache kann ungültig werden.



Schlüssel	Typ	Standardwert	Beschreibung
<code>mq.queues.default.syncpoint</code>	boolesch	false	Definiert das Standardverhalten für MQ PUT-Befehle, wenn weder <code>MQPMO_SYNCPOINT</code> noch <code>MQPMO_NO_SYNCPOINT</code> gesetzt sind. Wenn es auf <code>true</code> gesetzt ist, verhält es sich so, als ob Nachrichten während des PUT-Befehls NICHT direkt festgeschrieben werden. <code>MQPMO_SYNCPOINT</code> Wenn der Wert auf <code>False</code> gesetzt ist, verhält es sich so, als <code>MQPMO_NO_SYNCPOINT</code> ob Nachrichten während des PUT-Befehls direkt festgeschrieben werden.

Schlüssel	Typ	Standardwert	Beschreibung
<code>dataSimplifier.byteRangeBoundsCheck</code>	boolesch	false	Wenn der Wert auf true gesetzt ist, ByteRange wird sichergestellt, dass keine Daten mit falschen Werten erstellt werden. Der Standardwert lautet „false“.
<code>file.stdoutIntoLogger</code>	boolesch	false	Gibt an, ob in den SYSPUNCH Standarddateien SYSPRINT und nicht in den Standardausgabestream des Systems das Schreiben in den Logger aktiviert werden soll.
<code>tempFilesDirectory</code>	Zeichenfolge	Null	Gibt den Namen des Speicherorts der temporären Dateien an, die generiert werden.

Schlüssel	Typ	Standardwert	Beschreibung
tempFolderPattern	Zeichenfolge	Null	<p>Gibt ein Muster an, das verwendet wird, um den Namen des temporären Ordners auf der Grundlage der folgenden vordefinierten und anpassbaren Informationen dynamisch zu erstellen.</p> <p>HOST: der Hostname.</p> <p>JOBID: Die ID des Jobs.</p> <p>HASHCODE: Der Hashcode des Job-Kontextes.</p> <p>TIMESTAMP: das Muster, das beim Abrufen des Zeitstempels verwendet werden soll. Der Zielname des temporären Ordners ist TMP_DIR_{temp}. FolderPattern Im Fall des folgenden Musters beginnt der Name beispielsweise mit der Job-ID und endet mit dem „Zeitstempel“ tempFolderPattern:</p>

Schlüssel	Typ	Standardwert	Beschreibung
			JOBID, HOST=XXXXX, HASHCODE, timestamp =yyyymmddhhmmss. Wenn die Eigenschaft nicht zur YAML-Datei hinzugefügt wurde oder leer tempFolderPattern ist, lautet der Name des temporären Ordners „TMP_DIR_“ + this.hashCode () (Kontext). DefaultJob
database.cursor.raise.already.opened.error	boolesch	false	Gibt an, ob das Auslösen des SQLCODE-Fehlers 502 aktiviert werden soll, wenn ein bereits geöffneter Cursor geöffnet wird.
jics.spool.smtp.hostname	Zeichenfolge	Null	Gibt den SMTP-Serverhost an. Beispiel: smtp.xxx.com
jics.spool.smtp.port	Zeichenfolge	Null	Gibt den SMTP-Serverport an. Beispiel: 25
jics.spool.smtp.password	Zeichenfolge	Null	Gibt das Login-Passwort des SMTP-Servers an.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.spool.smtp.username</code>	Zeichenfolge	Null	Gibt den Benutzernamen des SMTP-Servers an.
<code>jics.spool.smtp.debug</code>	boolesch	false	Gibt den Debug-Modus für den SMTP-Server an.
<code>gapwalk-application.security</code>	Zeichenfolge	disabled	Schaltet die globale Sicherheitskonfiguration um (XSS, CORS, CSRF, OAUTH-Authentifizierung...). Zulässige Werte sind <code>disabled</code> und <code>enabled</code> .
<code>gapwalk-application.identity</code>	Zeichenfolge	Null	Globale Authentifizierungsmethode. Der empfohlene Wert ist <code>oauth</code> . Zulässige Werte sind <code>json</code> und <code>oauth</code> . Diese Option ist erforderlich, wenn sie <code>gapwalk-application.security</code> <code>enabled</code> ist.

Schlüssel	Typ	Standardwert	Beschreibung
<code>gapwalk-application.security.issuerUri</code>	Zeichenfolge	Null	Die Aussteller-URI des Identitätsanbieters (IdP). Diese Option ist erforderlich, wenn sie <code>gapwalk-application.identity</code> ist. <code>oauth</code>
<code>gapwalk-application.security.allowedOrigins</code>	Zeichenfolge []	null	Die Liste der Ursprünge, die zugelassen werden sollen. Diese Option <code>gapwalk-application.identity</code> muss auf <code>ingest</code> eingestellt sein <code>oauth</code> .
<code>gapwalk-application.security.claimGroupName</code>	Zeichenfolge	<code>cognito:groups</code>	Das Claim-Attribut, das die Liste aller Gruppen enthält, zu denen ein Benutzer gehört. Verwenden Sie <code>cognito:groups</code> für Amazon Cognito oder eine andere Zeichenfolge für einen ausländischen IdP.

Schlüssel	Typ	Standardwert	Beschreibung
<code>gapwalk-application.security.userName</code>	Zeichenfolge	username	Der Name des Anspruchsattributs, der zur Identifizierung einer Benutzeranfrage verwendet wird. Verwenden Sie <code>username</code> für Amazon Cognito, <code>preferred_username</code> für Keycloak oder eine andere Zeichenfolge für einen ausländischen IdP.
<code>gapwalk-application.localhostWhitelistingEnabled</code>	boolesch	true	Gibt an, ob die Authentifizierung anhand beliebiger Anfragen aktiviert werden soll. localhost
<code>gapwalk-application.defaultSuperAdminUserName</code>	Zeichenfolge	sadmin	Wenn deaktiviert <code>gapwalk-application.security</code> ist, gibt den lokalen Standard-Superbenutzernamen an.

Schlüssel	Typ	Standardwert	Beschreibung
<code>gapwalk-application.defaultSuperAdminUserPwd</code>	Zeichenfolge	<code>sadmin</code>	Wenn deaktiviert <code>gapwalk-application.security</code> ist, gibt das standardmäßige lokale Superuser-Passwort an.
<code>gapwalk-application.security.filterURIs</code>	Zeichenfolge	<code>disabled</code>	Schaltet die Konfiguration der Filter-URIs um. Zulässige Werte sind <code>disabled</code> und <code>enabled</code> .
<code>gapwalk-application.security.blockedURIs</code>	Zeichenfolge []	<code>null</code>	Die Liste der zu blockierenden URIs. Diese Option ist erforderlich, wenn sie <code>gapwalk-application.security.filterURIs</code> ist <code>enabled</code> .
<code>jics.redis.database</code>	Zahl	<code>0</code>	Gibt den Datenbank index für die Redis-Server Connection Factory im Bereich von 0 bis 15 an. Der Standardwert ist 0.
<code>jics.redis.maxTotal</code>	Zahl	<code>32</code>	Maximale Anzahl aktiver Verbindungen im Redis-Pool.



Schlüssel	Typ	Standardwert	Beschreibung
<code>jics.redis.maxIdle</code>	Zahl	32	Redis-Pool: maximale Anzahl inaktiver Verbindungen.
<code>jics.redis.minIdle</code>	Zahl	8	Redis-Pool-Mindestanzahl inaktiver Verbindungen.

## Verfügbare Eigenschaften für optionale Webanwendungen

Abhängig von Ihrer modernisierten Anwendung müssen Sie möglicherweise eine oder mehrere optionale Webanwendungen konfigurieren, die Abhängigkeiten wie z/OS, AS/400 oder IMS/MFS unterstützen. Die folgenden Tabellen enthalten Listen der verfügbaren Schlüssel-/Wertparameter für die Konfiguration der einzelnen optionalen Webanwendungen.

`gapwalk-utility-pgm.war`

Diese optionale Webanwendung enthält Unterstützung für Z/OS-Hilfsprogramme.

Diese Tabelle bietet einen umfassenden Überblick über die Schlüssel-/Wertparameter für diese Anwendung.

Schlüssel	Typ	Standardwert	Beschreibung
<code>logging.config</code>	Pfad	Klassenpfad: <code>logback-utility.xml</code>	Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar.
<code>spring.jta.enabled</code>	boolesch	<code>false</code>	Standardschlüssel. Wenn der Datenquellenunterstützungsmodus nicht <code>static-xa</code>

Schlüssel	Typ	Standardwert	Beschreibung
			ist, muss die auto Konfiguration von Spring JTA-Transaktionen deaktiviert werden.
spring.datasource.primary.jndi-name	Zeichenfolge	jdbc/primary	Der JNDI-Name (Java Naming And Directory Interface) für die primäre Datenquelle, wenn JNDI verwendet wird.
primary.datasource -driver-class-name -url -username -password	Standard-Spring-Datenquelle mit Unterschlüsseln		Enthält die Verbindungsinformationen für die Anwendungsdatenbank, sofern JNDI nicht verwendet wird. Muss dieselbe Konfiguration wie in der modernisierten YAML-Datei der Anwendung haben.  Alternativ wird die Verwendung von AWS-Geheimnissen dringend empfohlen, wie in <a href="#">xref: erklärt. / configuration/configuration.adoc</a> [Konfiguration].

Schlüssel	Typ	Standardwert	Beschreibung
encoding	Zeichenfolge	ASCII	Die in Hilfsprogrammen verwendete Kodierung. Erwartet eine gültige Kodierung CP1047IBM930,,ASCII,UTF-8
sysPunchEncoding	Zeichenfolge	ASCII	Der Zeichensatz für die Syspunch-Kodierung. Erwartet eine gültige Kodierung CP1047,,IBM930,... ASCII UTF-8
zonedMode	Zeichenfolge	EBCDIC_STRICT	Der Modus für die Kodierung oder Dekodierung von Datentypen in Zonen. Zulässige Werte sind EBCDIC_STRICT //EBCDIC_MODIFIED .AS400
unload.chunkSize	Zahl	0	Chunk-Größe, die für das Entladeprogramm verwendet wird.

Schlüssel	Typ	Standardwert	Beschreibung
<code>unload.sqlCodePointShift</code>	Zahl	0	Das SQL-Code-Hilfsprogramm Pointshift for Unload. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank von DB2 Postgresql ist.
<code>unload.columnFiller</code>	Zeichenfolge	Leerzeichen	Der Spaltenfüller des Hilfsprogramms zum Entladen.
<code>unload.variableCharIsNull</code>	boolesch	false	Verwenden Sie diesen Parameter im INFTILB-Programm. Wenn er auf gesetzt ist, <code>true</code> wird für alle Felder, die keine NULL-Werte zulassen, mit leeren Werten (Leerzeichen) eine leere Zeichenfolge zurückgegeben.
<code>unload.useDatabaseConfiguration</code>	boolesch	false	Gibt an, ob die Datums- oder Uhrzeitkonfiguration aus <code>application-main.yml</code> im Unload-Hilfsprogramm verwendet werden soll.

Schlüssel	Typ	Standardwert	Beschreibung
<code>unload.format.date</code>	Zeichenfolge	MM/dd/yyyy	Falls aktiviert, <code>unload.us eDatabase Configuration</code> das Datumsformat, das im Entladeprogramm verwendet werden soll.
<code>unload.format.time</code>	Zeichenfolge	Hh.mm.ss	Falls aktiviert, <code>unload.us eDatabase Configuration</code> das Zeitformat, das im Entladeprogramm verwendet werden soll.
<code>unload.format.timestamp</code>	Zeichenfolge	yyyy-MM-dd-hh.mm.ss.ssssss	Falls aktiviert, das Zeitstempelformat, das im Entladeprogramm verwendet werden soll. <code>unload.us eDatabase Configuration</code>
<code>unload.nbi.whenNull</code>	hexadezimal	6F	Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank Null ist.

Schlüssel	Typ	Standardwert	Beschreibung
<code>unload.nb i.whenNotNull</code>	hexadezimal	00	Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank nicht Null ist.
<code>unload.nb i.writeNullIndicator</code>	boolesch	false	Gibt an, ob der Null-Indikator in die Entladeausgabedate <code>i</code> geschrieben werden soll.
<code>unload.fetchSize</code>	Zahl	0	Ermöglicht es Ihnen, die Abrufgröße bei der Bearbeitung von Cursors im Entladeprogramm einzustellen.
<code>treatLargeNumberAs Integer</code>	boolesch	false	Gibt an, ob große Zahlen als behandelt werden sollen. Integer Sie werden <code>BigDecimal</code> standardmäßig wie behandelt.
<code>load.batchSize</code>	Zahl	0	Die Batchgröße des Ladedienstprogramms.
<code>load.format.localDate</code>	Zeichenfolge	<code>dd.mm.yyyy\ dd/MM/YYYY\ yyyy-mm-dd</code>	Das zu verwendende lokale Datumsformat des Load Utilitys.

Schlüssel	Typ	Standardwert	Beschreibung
<code>load.form at.localTime</code>	Zeichenfolge	hh:mm:ss\ hh.mm.ss	Das zu verwendende lokale Zeitformat des Load Utility.
<code>load.form at.dbDate</code>	Zeichenfolge	yyyy-MM-dd	Das zu verwendende Load Utility-Datenbankformat.
<code>load.form at.dbTime</code>	Zeichenfolge	hh:mm:ss	Die zu verwendende Uhrzeit der Load Utility-Datenbank.
<code>load.sqlC odePointShift</code>	Zahl	0s	Das SQL-Code-Hilfsprogramm Pointshift für Load. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank von DB2 Postgresql ist.
<code>forcedDate</code>	Zeichenfolge		Erzwingt das Datum auf das angegebene Datum, falls es eines gibt.
<code>frozenDate</code>	boolesch	true	Gibt an, ob das Datum eingefroren werden soll. Gilt nur, wenn <code>forcedDate</code> es auch gesetzt ist.

Schlüssel	Typ	Standardwert	Beschreibung
<code>jcl.type</code>	Zeichenfolge	<code>mvs</code>	.jcl-Dateityp. Zulässige Werte sind/ <code>jcl.vse</code> Die PRINT/REPRO-Befehle des IDCAMS-Hilfsprogramms geben 4 zurück, wenn die Datei für JCL-Dateien, die nicht mit VSE verknüpft sind, leer ist.
<code>hasGraphic</code>	boolesch	<code>false</code>	Ob das INFUTILB-Hilfsprogramm GRAPHIC DB2-Spalten verarbeiten muss.

### gapwalk-cl-command.war

Diese optionale Webanwendung enthält Unterstützung für AS/400-Hilfsprogramme.

Diese Tabelle bietet einen umfassenden Überblick über die Schlüssel-/Werteparameter für diese Anwendung.

Schlüssel	Typ	Standardwert	Beschreibung
<code>logging.config</code>	Pfad	Klassenpfad: <code>logback-utility.xml</code>	Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar.



Schlüssel	Typ	Standardwert	Beschreibung
<code>spring.jta.enabled</code>	boolesch	false	Standardschlüssel. Wenn der Datenquellenunterstützungsmodus nicht <code>static-xa</code> ist, muss die auto Konfiguration von Spring JTA-Transaktionen deaktiviert werden.
<code>spring.datasource.primary.jndi-name</code>	Zeichenfolge	<code>jdbc/primary</code>	Der JNDI-Name (Java Naming And Directory Interface) für die primäre Datenquelle, wenn JNDI verwendet wird.

Schlüssel	Typ	Standardwert	Beschreibung
<pre>primary.datasource + -driver-class-name + -url + -username + - password</pre>	Standard-Spring-Datenquelle mit Unterschlüsseln		<p>Enthält die Verbindungsinformationen für die Anwendungsdatenbank, sofern JNDI nicht verwendet wird. Muss dieselbe Konfiguration wie in der modernisierten YML-Datei der Anwendung haben.</p> <p>Alternativ wird die Verwendung von AWS-Geheimnissen dringend empfohlen, wie in <a href="#">xref: erklärt. / configuration/configuration.adoc [Konfiguration]</a>.</p>
encoding	Zeichenfolge	ASCII	Die in Hilfsprogrammen verwendete Kodierung. Erwartet eine gültige Kodierung CP1047IBM930,,ASCII,UTF-8
zonedMode	Zeichenfolge	EBCDIC_STRICT	Der Modus für die Kodierung oder Dekodierung von Datentypen in Zonen. Zulässige Werte sind EBCDIC_STRICT //EBCDIC_MODIFIED .AS400

Schlüssel	Typ	Standardwert	Beschreibung
commands-off	Zeichenfolge		Liste der auszuscha- ltenden Befehle, getrennt durch Kommas. Zulässige Werte sind PGM_BASIC RCVMSG,SNDRCVF,CHGVAR,Q DLTOVRCPYF,SNDDTAQ. Nützlich, wenn Sie ein vorhandenes Programm deaktivie- ren oder überschre- iben möchten. PGM_BASIC ist ein spezielles AWS Blu Age Runtime- Programm, das für Debugging-Zwecke entwickelt wurde.
forcedDate	Zeichenfolge		Setzt das Datum auf das angegebene Datum zurück, falls es eines gibt.

gapwalk-hierarchical-support.war

Diese optionale Webanwendung unterstützt IMS/MFS-Transaktionen.

Diese Tabelle bietet einen umfassenden Überblick über die Schlüssel-/Werteparameter für diese Anwendung.

Schlüssel	Typ	Standardwert	Beschreibung
<code>logging.config</code>	Pfad	Klassenpfad:logback-utility.xml	Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar.
<code>spring.jta.enabled</code>	boolesch	false	Standardschlüssel. Wenn der Datenquellenunterstützungsmodus nicht static-xa ist, muss die auto Konfiguration von Spring JTA-Transaktionen deaktiviert werden.
<code>jhdb.configuration.context.encoding</code>	Zeichenfolge		Die JHDB-Kodierung (Java Hierarchi cal Database). Erwartet eine gültige KodierungszeichenfolgeCP1047,,,... IBM930 ASCII UTF-8
<code>jhdb.checkpointPersistence</code>	Zeichenfolge	Keine	Der Checkpoint-Persistenzmodus. Zulässige Werte sindnone//add. end Wird verwendetadd, um Checkpoints beizubehalten, wenn ein neuer erstellt und

Schlüssel	Typ	Standardwert	Beschreibung
			<p>der Registrierung hinzugefügt wird. Wird verwendet, end um Checkpoints beim Herunterfahren des Servers beizubehalten. Alle anderen Werte deaktivieren die Persistenz. Beachten Sie, dass jedes Mal, wenn ein neuer Checkpoint zur Registrierung hinzugefügt wird, alle vorhandenen Checkpoints serialisiert und die Datei gelöscht wird. Es handelt sich nicht um eine Anlage an die vorhandenen Daten in der Datei. Je nach Anzahl der Checkpoints kann es also einige Auswirkungen auf die Leistung haben.</p>

## Konfiguration der Sicherheit für Gapwalk-Anwendungen

In den folgenden Themen wird beschrieben, wie Gapwalk-Anwendungen gesichert werden.

Es liegt in Ihrer Verantwortung, die richtige Konfiguration bereitzustellen, um sicherzustellen, dass die Verwendung des AWS Blu Age-Frameworks sicher ist.

Alle sicherheitsrelevanten Funktionen sind standardmäßig deaktiviert. Um die Authentifizierung (und CSRF, XSS, CSP usw.) zu aktivieren, stellen Sie auf und auf ein. `gapwalk-application.security.enabled` `gapwalk-application.security.identity.oauth`

## Themen

- [Konfiguration der URI-Zugänglichkeit für Gapwalk-Anwendungen](#)
- [Konfigurieren Sie die Authentifizierung für Gapwalk-Anwendungen](#)

## Konfiguration der URI-Zugänglichkeit für Gapwalk-Anwendungen

In diesem Thema wird beschrieben, wie die Filterung von URIs für Gapwalk-Anwendungen konfiguriert wird. Es ist kein Identitätsanbieter (IdP) erforderlich.

Um eine Liste von URIs zu blockieren, fügen Sie `application-main.yml` der Ihrer modernisierten Anwendung die folgenden beiden Zeilen hinzu und ersetzen Sie `URI-1`, `URI-2` usw. durch die URIs, die Sie blockieren möchten.

```
gapwalk-application.security.filterURIs: enabled
gapwalk-application.security.blockedURIs: URI-1, URI-2, URI-3
```

## Konfigurieren Sie die Authentifizierung für Gapwalk-Anwendungen

In den folgenden Themen wird beschrieben, wie die OAuth2-Authentifizierung für Gapwalk-Anwendungen mithilfe eines Identitätsanbieters (IdP) wie Cognito oder Keycloak konfiguriert wird.

## Themen

- [Gapwalk OAuth2-Authentifizierung mit Amazon Cognito](#)
- [Gapwalk OAuth2-Authentifizierung mit Keycloak](#)

## Gapwalk OAuth2-Authentifizierung mit Amazon Cognito

In diesem Thema wird beschrieben, wie die OAuth2-Authentifizierung für Gapwalk-Anwendungen mit Amazon Cognito als Identitätsanbieter (IdP) konfiguriert wird.

## Voraussetzungen

In diesem Tutorial werden wir Amazon Cognito als IdP und PlanetDemo als modernisiertes Projekt verwenden.

Sie können jeden anderen externen Identitätsanbieter verwenden. Die ClientRegistration Informationen müssen von Ihrem IdP bezogen werden und sind für die Gapwalk-Authentifizierung erforderlich. Weitere Informationen finden Sie im [Amazon Cognito Entwicklerhandbuch](#).

Die Informationen: ClientRegistration

Kunden-ID

Die ID der ClientRegistration. In unserem Beispiel wird es so sein. PlanetsDemo

Kundengeheimnis

Das Geheimnis Ihres Kunden.

Endpunkt der Autorisierung

Der Autorisierungsendpunkt-URI für den Autorisierungsserver.

Token-Endpunkt

Der Token-Endpunkt-URI für den Autorisierungsserver.

Jwks-Endpunkt


Der URI, der zum Abrufen des JSON-Webschlüssels (JWK) verwendet wird, der die Schlüssel für die Validierung der vom Autorisierungsserver ausgestellten JSON-Websignatur enthält.

URI umleiten

Der URI, zu dem der Autorisierungsserver den Endbenutzer weiterleitet, wenn der Zugriff gewährt wird.

Einrichtung von Amazon Cognito

Zunächst erstellen und konfigurieren wir einen Amazon Cognito Cognito-Benutzerpool und einen Benutzer, die wir mit unserer bereitgestellten Gapwalk-Anwendung zu Testzwecken verwenden werden.

 Note

Wenn Sie einen anderen IdP verwenden, können Sie diesen Schritt überspringen.

## Benutzerpool erstellen

1. Gehen Sie zu Amazon Cognito in AWS Management Console und authentifizieren Sie sich mit Ihren AWS Anmeldeinformationen.
2. Wählen Sie User Pools (Benutzerpools) aus.
3. Wählen Sie Create a user pool.
4. Behalten Sie unter Anmeldeerlebnis konfigurieren den Standardanbietertyp für den Cognito-Benutzerpool bei. Sie können eine oder mehrere Anmeldeoptionen für den Cognito-Benutzerpool auswählen. Wählen Sie zunächst Benutzername und dann Weiter.

[Amazon Cognito](#) > [User pools](#) > Create user pool

The screenshot shows the 'Configure sign-in experience' step in the AWS Management Console. On the left, a vertical progress bar lists six steps: Step 1 (selected), Step 2, Step 3, Step 4, Step 5, and Step 6. The main content area is titled 'Configure sign-in experience' and includes a sub-header 'Authentication providers'. Below this, there are two provider type options: 'Cognito user pool' (selected) and 'Federated identity providers'. The 'Cognito user pool' option is highlighted with a blue border. Below the provider types, there are 'Cognito user pool sign-in options' and 'User name requirements' sections. A yellow warning box at the bottom states: 'Cognito user pool sign-in options can't be changed after the user pool has been created.' At the bottom right, there are 'Cancel' and 'Next' buttons.

Step 1  
● **Configure sign-in experience**

Step 2  
○ Configure security requirements

Step 3  
○ Configure sign-up experience

Step 4  
○ Configure message delivery

Step 5  
○ Integrate your app

Step 6  
○ Review and create

### Configure sign-in experience [Info](#)

Your app users can sign in to your user pool with a user name and password, or sign in with a third-party identity provider.

#### Authentication providers

Configure the providers that are available to users when they sign in.

#### Provider types

Choose whether users will sign in to your Cognito user pool, a federated identity provider, or both. Amazon Cognito has different pricing for federated users and user pool users. [Learn more about pricing](#)

- Cognito user pool**  
Users can sign in using their email address, phone number, or user name. User attributes, group memberships, and security settings will be stored and configured in your user pool.
- Federated identity providers**  
Users can sign in using credentials from social identity providers like Facebook, Google, Amazon, and Apple; or using credentials from external directories through SAML or Open ID Connect. You can manage user attribute mappings and security for federated users in your user pool.

#### Cognito user pool sign-in options [Info](#)

Choose the attributes in your user pool that are used to sign in. If you select only one attribute, or you select a user name and at least one other attribute, your user can sign in with all of the selected options. If you select only phone number and email, your user will be prompted to select one of the two sign-in options when they sign up.

- User name**
- Email
- Phone number

#### User name requirements

- Allow users to sign in with a preferred user name
- Make user name case sensitive

**⚠ Cognito user pool sign-in options can't be changed after the user pool has been created.**

[Cancel](#) [Next](#)

5. Behalten Sie unter Sicherheitsanforderungen konfigurieren die Standardeinstellungen bei und deaktivieren Sie die Multi-Faktor-Authentifizierung, indem Sie Kein MFA und dann Weiter auswählen.



Advanced security features can protect your production user accounts from malicious sign-in attempts. Activate it today from [App Integration](#). [Learn more](#)

**Configure security requirements**

Step 2  
Configure sign-up experience

Step 4  
Configure message delivery

Step 5  
Integrate your app

Step 6  
Review and create

**Password policy** [Info](#)  
Create a password policy to define the length and complexity of the passwords your users can set.

**Password policy mode** [Info](#)

**Cognito defaults**  
Use default password requirements.

**Custom**  
Use password requirements that you define.

**Password minimum length**  
8 character(s)

**Password requirements**  
Contains at least 1 number  
Contains at least 1 special character  
Contains at least 1 uppercase letter  
Contains at least 1 lowercase letter

**Temporary passwords set by administrators expire in**  
7 day(s)

**Multi-factor authentication**  
Configure secure access to your app by enforcing multi-factor authentication (MFA) during the user sign-in process. MFA settings are applied to all app clients.

**MFA enforcement** [Info](#)

**Require MFA - Recommended**  
Users must provide an additional authentication factor when signing in.

**Optional MFA**  
Users can sign in with a single authentication factor, and can choose to add additional authentication factors.

**No MFA**  
Users can only sign in with a single authentication factor. This is the least secure option.

**User account recovery**  
Configure how users will recover their account when they forget their password. Recipient message and data rates apply.

**Self-service account recovery** [Info](#)

**Enable self-service account recovery - Recommended**  
Allow forgot-password operations in your user pool. In the hosted UI sign-in page, a "Forgot your password?" link is displayed. When this feature is not enabled, administrators reset passwords with the Cognito API.

**Delivery method for user account recovery messages** [Info](#)

Select how your user pool will deliver messages when users request an account recovery code. SMS messages are charged separately by Amazon SNS. Email messages are charged separately by Amazon SES. [Learn more about pricing](#)

**Email only**

**SMS only**

**Email if available, otherwise SMS**

**SMS if available, otherwise email**

**SMS if available, otherwise email, and allow a user to reset their password via SMS if they are also using it for MFA**

Cancel [Previous](#) [Next](#)

6. Deaktivieren Sie aus Sicherheitsgründen die Option Selbstregistrierung aktivieren und wählen Sie dann Weiter aus.

**Self-service sign-up** [Info](#)  
Choose whether new users of your app can register for an account themselves.

**Self-registration** [Info](#)

**Enable self-registration**  
Display a "Sign up" link on the sign-in page in the hosted UI, and allow the use of public APIs to create new user accounts. When this feature is not enabled, federation and administrative API operations create user profiles.

7. Wählen Sie E-Mail mit Cognito senden und dann Weiter.



## Email

Configure how your user pool sends email messages to users.

Email provider [Info](#)



**Send email with Amazon SES - Recommended**  
Send emails using an Amazon SES verified identity in your account. We recommend this option for higher email volume and production workloads.

**Send email with Cognito**  
Use Cognito's default email address as a temporary start for development. You can use it to send up to 50 emails a day.

You must have configured a verified sender with [Amazon SES](#)  to use the SES feature. [Learn more](#) 

SES Region [Info](#)  
Europe (Ireland)

FROM email address [Info](#)  
By default "no-reply@verificationemail.com" will be used. You can also choose a different email address that you have previously verified with Amazon SES.

no-reply@verificationemail.com  

REPLY-TO email address - *optional* [Info](#)  
If you set an invalid reply-to address, sending restrictions may be imposed on your account.

8. Geben Sie unter Integrieren Sie Ihre App einen Namen für Ihren Benutzerpool an. Wählen Sie unter Gehostete Authentifizierungsseiten die Option Cognito Hosted UI verwenden aus.

Advanced security features can protect your production user accounts from malicious sign-in attempts. Activate it today from [App Integration](#). [Learn more](#)

Amazon Cognito > User pools > Create user pool

Step 1  
Configure sign-in experience

Step 2  
Configure security requirements

Step 3  
Configure sign-up experience

Step 4  
Configure message delivery

**Step 5  
Integrate your app**

Step 6  
Review and create

### Integrate your app Info

Set up app integration for your user pool with Cognito's built-in authentication and authorization flows.

**User pool name**  
Create a friendly name for your user pool.

**User pool name**

User pool names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + - . @ -

⚠
Your user pool name can't be changed once this user pool is created.

**Hosted authentication pages**

Choose whether to use Cognito's Hosted UI and OAuth 2.0 server for user sign-up and sign-in flows.

**Use the Cognito Hosted UI**  
Build hosted sign-up, sign-in, and OAuth 2.0 service endpoints in Amazon Cognito. When this feature is not enabled, use Cognito API operations to perform sign-up and sign-in.

**Domain** Info

Configure a domain for your Hosted UI and OAuth 2.0 endpoints. To use the Hosted UI, you must choose a domain where authentication endpoints will be created.

**Domain type**

**Use a Cognito domain**  
Enter an identifying prefix to use in an Amazon-owned domain. For production apps, we recommend using a custom domain instead.

**Use a custom domain**  
Enter a domain that you own for Cognito-hosted sign-up and sign-in pages. You must provide a DNS record and an AWS Certificate Manager (ACM) certificate to use a custom domain. We recommend using a custom domain for production workloads.

**Cognito domain**

Enter a domain prefix.

Domain prefixes may only include lowercase, alphanumeric characters, and hyphens. You can't use the text aws, amazon, or cognito in the domain prefix. Your domain prefix must be unique within the current Region.

✔ Available

**Initial app client**

Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

**App type** Info

Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

9. Der Einfachheit halber wählen Sie unter Domain die Option Eine Cognito-Domäne verwenden aus und geben Sie ein Domainpräfix ein, `https://planetsdemo` zum Beispiel. Die Demo-App muss als Client hinzugefügt werden.

1. Wählen Sie unter Anfänglicher App-Client die Option Vertraulicher Client aus. Geben Sie einen App-Clientnamen ein, z. B. **planetsdemo**, und wählen Sie dann Generate a Client Secret aus.
2. Geben Sie im Feld Zulässige Rückruf-URL die URL ein, zu der der Benutzer nach der Authentifizierung weitergeleitet werden soll. Die URL muss mit `/login/oauth2/code/cognito` enden. Zum Beispiel für unsere Anwendung und Backend-Gapwalk- und BAC-Anwendungen:

```

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

```

Sie können die URL später bearbeiten.

### Initial app client

Configure an app client. App clients are single-app platforms in your user pool that have permissions to call unauthenticated API operations. A user pool can have multiple app clients.

**App type** [Info](#)  
Select an app type and we will automatically populate common default settings. You can add additional app clients after the user pool is created.

**Public client**  
A native, browser or mobile-device app. Cognito API requests are made from user systems that are not trusted with a client secret.

**Confidential client**  
A server-side application that can securely store a client secret. Cognito API requests are made from a central server.

**Other**  
A custom app. Choose your own grant, auth flow, and client-secret settings.

**App client name** [Info](#)  
Enter a friendly name for your app client.

planetsdemo

App client names are limited to 128 characters or less. Names may only contain alphanumeric characters, spaces, and the following special characters: + = , @ -

**Client secret** [Info](#)  
Choose whether your app client will have a client secret. Client secrets are used by the server-side component of an app to authorize API requests. Using a client secret can prevent a third party from impersonating your client.

**Generate a client secret**  
 Don't generate a client secret

**You cannot change or remove a client secret after you allow Amazon Cognito to generate it for your app client.**

**Allowed callback URLs** [Info](#)  
Enter at least one callback URL to redirect the user back to after authentication. This is typically the URL for the app receiving the authorization code issued by Cognito. You may use HTTPS URLs, as well as custom URL schemes.

**URL**

http://localhost:8080/bac Remove

http://localhost:8080/bac/login/oauth2/code/cognito Remove

http://localhost:8080/gapwalk-application Remove

http://localhost:8080/gapwalk-application/login/oauth2/code/cognito Remove

http://localhost:8080/planetsdemo Remove

http://localhost:8080/planetsdemo/login/oauth2/code/cognito Remove

[Add another URL](#)  
You can add 94 more URLs.

**Advanced app client settings**

- Geben Sie unter Zulässige Abmelde-URLs die URL der Abmeldeseite ein, zu der Amazon Cognito weiterleiten soll, wenn Ihre Anwendung Benutzer abmeldet. Zum Beispiel für Backend-Anwendungen Gapwalk und BAC:

```
http://localhost:8080/bac/logout
http://localhost:8080/gapwalk-application/logout
http://localhost:8080/planetsdemo/logout
```

Sie können die URL später bearbeiten.

- Behalten Sie die Standardwerte in den Abschnitten Erweiterte App-Client-Einstellungen und Lese- und Schreibberechtigungen für Attribute bei.
- Wählen Sie Weiter aus.
- Überprüfen Sie unter Überprüfen und erstellen Ihre Auswahl und wählen Sie dann Benutzerpool erstellen aus.

Weitere Informationen finden Sie unter [Benutzerpool erstellen](#).

## Benutzererstellung

Da die Selbstregistrierung deaktiviert ist, erstellen Sie einen Amazon Cognito Cognito-Benutzer. Navigieren Sie zu Amazon Cognito in der AWS Management Console. Wählen Sie den Benutzerpool aus, den Sie erstellt haben, und klicken Sie dann unter Benutzer auf Benutzer erstellen.

Wählen Sie unter Benutzerinformationen die Option E-Mail-Einladung senden aus, geben Sie einen Benutzernamen und eine E-Mail-Adresse ein und wählen Sie Passwort generieren aus. Wählen Sie Create user (Benutzer erstellen) aus.

## Erstellung einer Rolle

Erstellen Sie auf der Registerkarte Gruppen 3 Gruppen (SUPER\_ADMIN, ADMIN und USER) und ordnen Sie Ihren Benutzer einer oder mehreren dieser Gruppen zu. Diese Rollen werden später von der Gapwalk-Anwendung ROLE\_SUPER\_ADMIN, ROLE\_ADMIN und ROLE\_USER zugeordnet, um den Zugriff auf einige eingeschränkte API-REST-Aufrufe zu ermöglichen.

## Integration von Amazon Cognito in die Gapwalk-Anwendung

Nachdem Ihr Amazon Cognito Cognito-Benutzerpool und Ihre Benutzer bereit sind, gehen Sie in die `application-main.yml` Datei Ihrer modernisierten Anwendung und fügen Sie den folgenden Code hinzu:

```
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>
gapwalk-application.security.localhostWhitelistingEnabled: false

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
jwks.json    jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/
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
```

Ersetzen Sie die folgenden Platzhalter wie beschrieben:

1. Gehen Sie zu Amazon Cognito in AWS Management Console und authentifizieren Sie sich mit Ihren AWS Anmeldeinformationen.
2. Wählen Sie Benutzerpools und wählen Sie den Benutzerpool aus, den Sie erstellt haben. Sie finden Ihre *Pool-ID unter* Benutzerpool-ID.
3. Wählen Sie App-Integration, wo Sie *Ihre Cognito-Domain* finden, und gehen Sie dann zu App-Clients und Analytics und wählen Sie Ihre App aus.
4. ***Im App-Client: YourApp finden Sie den Client-Namen, die Client-ID und das Client-Geheimnis (Client-Geheimnis anzeigen).***
5. *region-id entspricht der AWS Region-ID*, in der Sie Ihren Amazon Cognito Cognito-Benutzer und Ihren Benutzerpool erstellt haben. Beispiel: eu-west-3.
6. Geben Sie für die ***Umleitungs-URI die URI*** ein, die Sie für die zulässige Rückruf-URL angegeben haben. In unserem Beispiel ist es. `http://localhost:8080/planetsdemo/login/oauth2/code/cognito`

Sie können jetzt Ihre Gapwalk-Anwendung bereitstellen und den zuvor erstellten Benutzer verwenden, um sich bei Ihrer App anzumelden.

## Gapwalk OAuth2-Authentifizierung mit Keycloak

In diesem Thema wird beschrieben, wie die OAuth2-Authentifizierung für Gapwalk-Anwendungen mit Keycloak als Identitätsanbieter (IdP) konfiguriert wird. In diesem Tutorial verwenden wir Keycloak 24.0.0.

### Voraussetzungen

- [Keycloak](#)

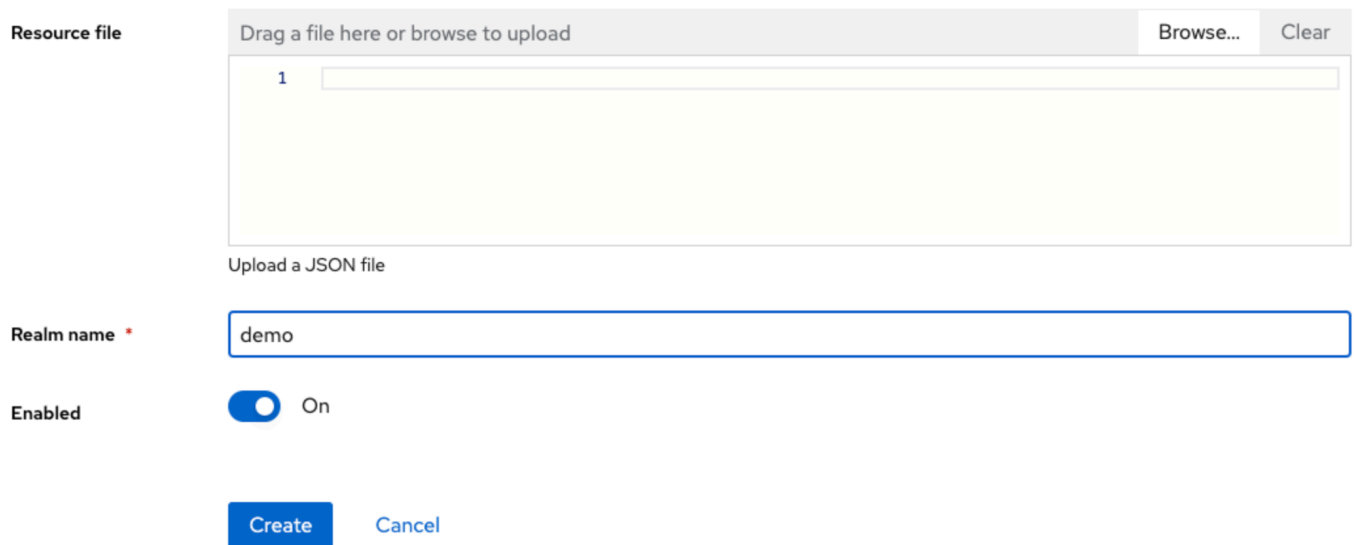
- Gapwalk-Anwendung

## Einrichtung von Keycloak

1. Gehen Sie in Ihrem Webbrowser zu Ihrem Keycloak-Dashboard. Die Standardanmeldedaten sind admin/admin. Gehen Sie zur Navigationsleiste oben links und erstellen Sie einen Bereich mit dem Namen **demo**, wie in der folgenden Abbildung gezeigt.

### Create realm

A realm manages a set of users, credentials, roles, and groups. A user belongs to and logs into a realm. Realms are isolated from one another and c



Resource file

Drag a file here or browse to upload

Browse... Clear

1

Upload a JSON file

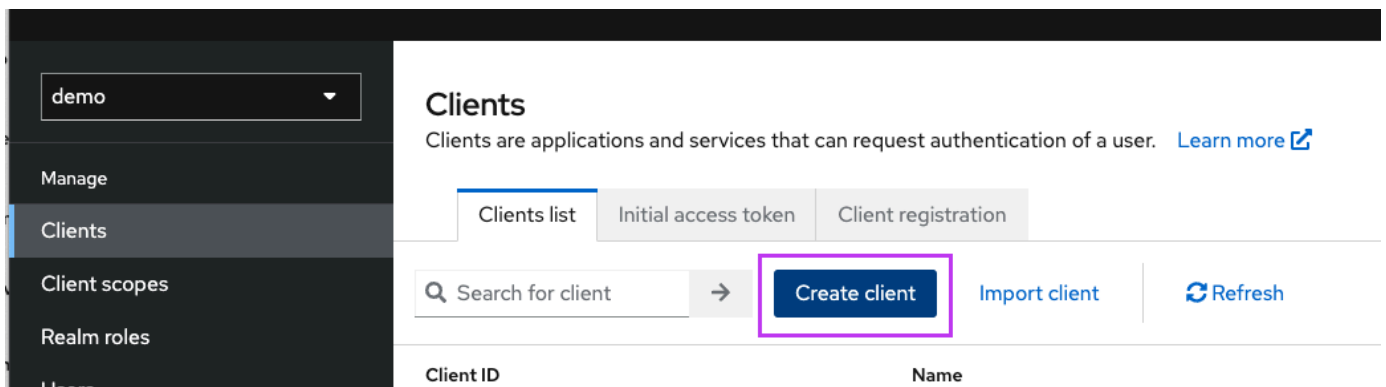
Realm name \*

demo

Enabled  On

Create Cancel

2. Erstellen Sie einen Client mit dem Namen **app-demo**.



demo

Manage

Clients

Client scopes

Realm roles

Users

## Clients

Clients are applications and services that can request authentication of a user. [Learn more](#)

Clients list Initial access token Client registration

Search for client → Create client Import client Refresh






Client ID	Name
-----------	------

localhost:8080 Ersetzen Sie durch die Adresse Ihrer Gapwalk-Anwendung

## General settings

Client ID * ?	<input type="text" value="app-demo"/>
Name ?	<input type="text"/>
Description ?	<input type="text"/>
Always display in UI ?	<input type="checkbox"/> Off

## Access settings

Root URL ?	<input type="text" value="http://localhost:8080"/>
Home URL ?	<input type="text"/>
Valid redirect URIs ?	<input type="text" value="http://localhost:8080/*"/>  <input type="text" value="https://localhost:8080/*"/>  <a href="#">+ Add valid redirect URIs</a>
Valid post logout redirect URIs ?	<input type="text" value="http://localhost:8080/*"/>  <input type="text" value="https://localhost:8080/*"/>  <a href="#">+ Add valid post logout redirect URIs</a>
Web origins ?	<input type="text" value="+"/>  <a href="#">+ Add web origins</a>



## Capability config

**Client authentication**  On

**Authorization**  Off

**Authentication flow**

- Standard flow [?](#)
- Implicit flow [?](#)
- OAuth 2.0 Device Authorization Grant [?](#)
- OIDC CIBA Grant [?](#)
- Direct access grants [?](#)
- Service accounts roles [?](#)

3. Um Ihr Client-Geheimnis zu erhalten, wählen Sie Clients, dann App-Demo und dann Credentials.

**app-demo** OpenID Connect  Enabled [?](#) Action ▾

Clients are applications and services that can request authentication of a user.

Settings | Keys | **Credentials** | Roles | Client scopes | Service accounts roles | Sessions | Advanced

**Client Authenticator** [?](#) Client Id and Secret

**Save**

**Client Secret** [?](#) 5wfK2WyAPQ2Sap732p2Jf39LitIDzYk 🗑️ 📄 **Regenerate**

4. Wählen Sie Clients, dann Client-Bereiche und dann Vordefinierten Mapper hinzufügen aus. Wählen Sie Realm-Rollen aus.

## Add predefined mappers

Choose any of the predefined mappings from this table

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	groups	Map a user realm role to a token claim.
<input checked="" type="checkbox"/>	realm roles	Map a user realm role to a token claim.

5. Bearbeiten Sie Ihre Realm-Rolle mit der in der folgenden Abbildung gezeigten Konfiguration.


[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  OnToken Claim Name 

keycloak:groups

Claim JSON Type 


String

Add to ID token  On

Add to access token

 On

Add to lightweight

access token  OnAdd to userinfo  On

Add to token

introspection  On

6. Merken Sie sich den definierten Namen für den Token-Anspruch. Sie benötigen diesen Wert in der Definition der Gapwalk-Einstellungen für die `gapwalk-application.security.claimGroupName` Eigenschaft.

The screenshot shows the 'Realm roles' configuration page. On the left, a dark sidebar contains a navigation menu with 'demo' at the top and 'Realm roles' selected. The main area has a header 'Realm roles' and a sub-header 'Realm roles are the roles that you define for use in the current realm. [Lea](#)'. Below the header is a search bar with the text 'Search role by name', a right arrow, a blue 'Create role' button, and a 'Refresh' button with a circular arrow icon. A table below lists the roles:

Role name
ADMIN
SADMIN
USER

7. Wählen Sie Realms-Rollen und erstellen Sie 3 Rollen: **SUPER\_ADMIN**, **ADMIN** und **USER**. Diese Rollen werden später der Gapwalk-Anwendung `ROLE_SUPER_ADMIN`, `ROLE_ADMIN`, und `ROLE_USER` zugeordnet, um auf einige eingeschränkte API-REST-Aufrufe zugreifen zu können.

The screenshot shows the 'User details' page. The left sidebar has 'demo' at the top and 'Users' selected. The main area has a breadcrumb 'Users > User details' and a header 'User'. Below the header are tabs for 'Details', 'Credentials', 'Role mapping', 'Groups', 'Consents', and 'Identity'. The 'Role mapping' tab is active. Below the tabs is a search bar with the text 'Search by name', a right arrow, a checked 'Hide inherited roles' checkbox, and a blue 'Assign role' button. A table below lists the roles with checkboxes:

<input type="checkbox"/> Name
<input type="checkbox"/> default-roles-demo
<input type="checkbox"/> USER
<input type="checkbox"/> ADMIN
<input type="checkbox"/> SADMIN

## Integration von Keycloak in die Gapwalk-Anwendung

Bearbeiten Sie Ihre wie folgt: `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
gapwalk-application.security.localhostWhitelistingEnabled: false

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
```

Ersetzen Sie `<KEYCLOAK_SERVER_HOSTNAME>`, `<YOUR_REALM_NAME>``<YOUR_CLIENT_ID>`, und `<YOUR_CLIENT_SECRET>` durch Ihren Keycloak-Server-Hostnamen, Ihren Realm-Namen, Ihre Client-ID und Ihr Client-Geheimnis.

## AWS Blu-Age-Laufzeit-APIs

Die AWS Blu Age Runtime verwendet mehrere Webanwendungen, um REST-Endpunkte verfügbar zu machen, und bietet so Möglichkeiten, mithilfe von REST-Clients mit den modernisierten Anwendungen zu interagieren (z. B. das Aufrufen von Jobs mithilfe eines Schedulers).

Der Zweck dieses Dokuments besteht darin, die verfügbaren REST-Endpunkte aufzulisten und Einzelheiten zu folgenden Themen zu enthalten:

- Ihre Rolle
- Die Art, sie richtig zu benutzen

Die Liste der Endgeräte ist je nach Art des bereitgestellten Dienstes und der Webanwendung, die die Endpunkte verfügbar macht, in Kategorien unterteilt.

Wir gehen davon aus, dass Sie bereits über Grundkenntnisse in der Verwendung von REST-Endpunkten mit speziellen Tools wie [POSTMAN](#), [Thunder Client](#), [CURL](#), Webbrowsern usw. verfügen. oder schreiben Sie Ihren eigenen Code, um einen API-Aufruf zu tätigen.

### Themen

- [URLs erstellen](#)
- [Gapwalk-Anwendung](#)
- [REST-Endpunkte der Blusam Application Console](#)
- [JICS-Anwendungskonsole](#)
- [Datenstrukturen](#)

## URLs erstellen

Jede der unten aufgeführten Webanwendungen definiert einen Stammpfad, der von allen Endpunkten gemeinsam genutzt wird. Jeder Endpunkt fügt dann seinen eigenen dedizierten Pfad hinzu. Die resultierende URL, die verwendet werden soll, ist das Ergebnis der Verkettung der Pfade. Wenn wir zum Beispiel den ersten Endpunkt für die Gapwalk-Anwendung betrachten, haben wir:

- `/gapwalk-application` für den Stammpfad der Webanwendung.
- `/scripts` für den dedizierten Endpunktpfad.

Die resultierende URL, die verwendet werden soll, lautet `http://server:port/gapwalk-application/scripts`

`server`

zeigt auf den Servernamen (den, der die angegebene Webanwendung hostet).

`port`

der vom Server bereitgestellte Port.

## Gapwalk-Anwendung

Endpunkte für die Gapwalk-Webanwendung verwenden den Root-Pfad. `/gapwalk-application`

Themen

- [Endgeräte im Zusammenhang mit Batch-Jobs \(modernisierte JCLs und ähnliches\)](#)
- [Metriken und Endpunkte](#)
- [Andere Endpunkte](#)
- [Endpunkte im Zusammenhang mit Job Queues](#)

### Endgeräte im Zusammenhang mit Batch-Jobs (modernisierte JCLs und ähnliches)

Batch-Jobs können entweder synchron oder asynchron ausgeführt werden (siehe Details unten). Batch-Jobs werden mithilfe von Groovy-Skripten ausgeführt, die das Ergebnis der Modernisierung von Legacy-Skripten (JCL) sind.

Themen

- [Listet die bereitgestellten Skripte auf](#)
- [Starten Sie ein Skript synchron](#)
- [Starten Sie ein Skript asynchron](#)
- [Auflisten ausgelöster Skripte](#)
- [Details zur Auftragsausführung werden abgerufen](#)


- [Listet asynchron gestartete Skripten auf, die beendet werden können](#)
- [Listet synchron gestartete Skripten auf, die beendet werden können](#)
- [Beenden einer bestimmten Jobausführung](#)
- [Auflisten vorhandener Checkpoints aus Gründen der Neustartfähigkeit](#)
- [Einen Job neu starten \(synchron\)](#)
- [Einen Job neu starten \(asynchron\)](#)
- [Einstellung des Thread-Limits für asynchrone Jobausführungen](#)

Listet die bereitgestellten Skripte auf

- Unterstützte Methode: GET
- Pfad: /scripts
- Argumente: keine
- Dieser Endpunkt gibt die Liste der bereitgestellten Groovy-Skripte auf dem Server als Zeichenfolge zurück. Dieser Endpunkt ist in erster Linie für die Verwendung in einem Webbrowser vorgesehen, da der resultierende String eine HTML-Seite mit aktiven Links ist (ein Link pro startbarem Skript — siehe Beispiel unten).

Beispielantwort:

```
<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

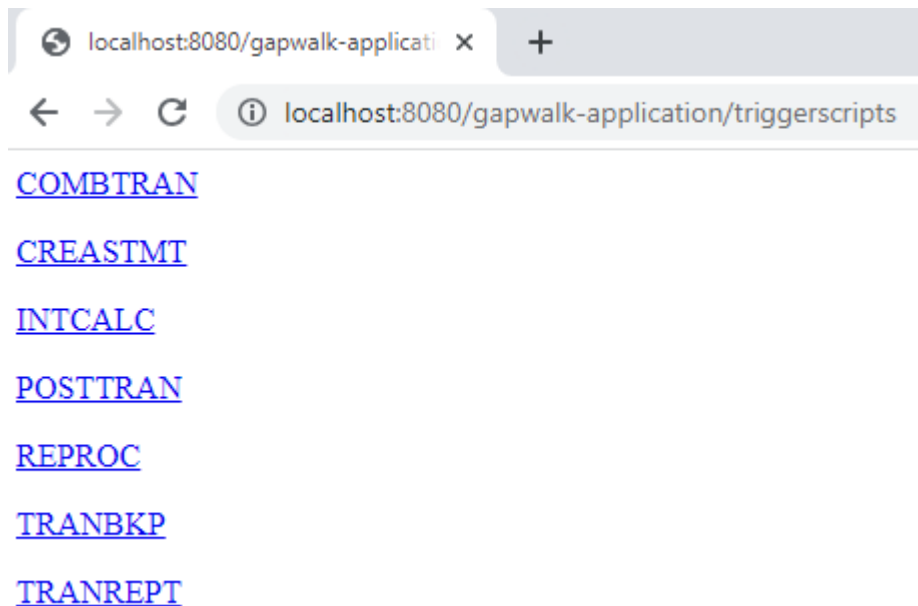
Die Links stellen die URL dar, die verwendet werden soll, um jedes aufgelistete Skript synchron zu starten.

- Unterstützte Methode: GET
- Pfad: /triggerscripts
- Argumente: keine



- Dieser Endpunkt gibt die Liste der bereitgestellten Groovy-Skripte auf dem Server als Zeichenfolge zurück. Dieser Endpunkt ist in erster Linie für die Verwendung in einem Webbrowser vorgesehen, da der resultierende String eine HTML-Seite mit aktiven Links ist (ein Link pro startbarem Skript — siehe Beispiel unten).

Im Gegensatz zur vorherigen Endpunktantwort stellen die Links die URL dar, die verwendet werden soll, um jedes aufgelistete Skript asynchron zu starten.



Starten Sie ein Skript synchron

Dieser Endpunkt hat zwei Varianten mit dedizierten Pfaden für die GET- und POST-Nutzung (siehe unten).

- Unterstützte Methode: GET
- Pfad: /script/{scriptId:..+}
- Unterstützte Methode: POST
- Pfad: /post/script/{scriptId:..+}
- Argumente:
  - Kennung des zu startenden Skripts
  - optional: Parameter, die unter Verwendung von Anforderungsparametern (als a bezeichnetMap<String, String>) an das Skript übergeben werden. Die angegebenen Parameter werden automatisch zu den [Bindungen](#) des aufgerufenen Groovy-Skripts hinzugefügt.

- Der Aufruf startet das Skript mit der angegebenen Kennung, verwendet zusätzliche Parameter, falls vorhanden, und wartet, bis die Ausführung des Skripts abgeschlossen ist, bevor eine Nachricht (String) zurückgegeben wird, die entweder:
  - „Erledigt.“ (wenn die Auftragsausführung reibungslos verlief).
  - Eine JSON-Fehlermeldung mit Details darüber, was bei der Jobausführung schief gelaufen ist. Weitere Details können aus den Serverprotokollen abgerufen werden, um zu verstehen, was bei der Auftragsausführung schief gelaufen ist.

```
{
  "exitCode": -1,
  "stepName": "STEP15",
  "program": "CBACT04C",
  "status": "Error"
}
```

Anhand der Serverprotokolle können wir feststellen, dass es sich um ein Bereitstellungsproblem handelt (das erwartete Programm wurde nicht ordnungsgemäß bereitgestellt, sodass es nicht gefunden werden kann, sodass die Auftragsausführung fehlschlägt):

```
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

Die synchronen Aufrufe sollten für Jobs mit kurzer Laufzeit reserviert werden. Jobs mit langer Laufzeit sollten lieber asynchron gestartet werden (siehe dedizierter Endpunkt unten).

Starten Sie ein Skript asynchron

- Unterstützte Methoden: GET/POST
- Pfad: /triggerscript/{scriptId:..+}
- Argumente:

- Kennung des zu startenden Skripts
- optional: Parameter, die unter Verwendung von Anforderungsparametern (als `bezeichnetMap<String, String>`) an das Skript übergeben werden. Die angegebenen Parameter werden automatisch zu <https://docs.groovy-lang.org/latest/html/api/groovy/lang/Binding.html#bindings> des aufgerufenen Groovy-Skripts hinzugefügt.
- Im Gegensatz zum obigen synchronen Modus wartet der Endpunkt nicht auf den Abschluss der Auftragsausführung, um eine Antwort zu senden. Die Auftragsausführung wird sofort gestartet, wenn dafür ein verfügbarer Thread gefunden wird, und es wird sofort eine Antwort an den Aufrufer gesendet, die die Auftragsausführungs-ID enthält, eine eindeutige Kennung, die die Auftragsausführung darstellt. Diese kann verwendet werden, um den Status der Auftragsausführung abzufragen oder das Abbrechen einer Auftragsausführung zu erzwingen, bei der eine Fehlfunktion vermutet wird. Das Format der Antwort ist:

```
Triggered script <script identifier> [unique job execution id] @ <date and time>
```

- Da die asynchrone Ausführung des Jobs auf einer festen, begrenzten Anzahl von Threads basiert, wird die Auftragsausführung möglicherweise nicht gestartet, wenn kein verfügbarer Thread gefunden werden konnte. In diesem Fall sieht die zurückgegebene Nachricht eher so aus:

```
Script [<script identifier>] NOT triggered - Thread limit reached (<actual thread limit>) - Please retry later or increase thread limit.
```

Im `settriggerthreadlimit` Endpunkt unten erfahren Sie, wie Sie das Thread-Limit erhöhen können.

Beispielantwort:

```
Triggered script INTCALC [d43cbf46-4255-4ce2-aac2-79137573a8b4] @ 06-12-2023 16:26:15
```

Die eindeutige ID für die Auftragsausführung ermöglicht es, bei Bedarf schnell zugehörige Protokolleinträge in den Serverprotokollen abzurufen. Sie wird auch von mehreren anderen Endpunkten verwendet, die weiter unten beschrieben werden.

### Auflisten ausgelöster Skripte

- Unterstützte Methoden: GET

- Pfade: /triggeredscripts/{status:.+}, /triggeredscripts/{status:.+}/  
{namefilter}
- Argumente:
  - Status (verpflichtend): Der Status der ausgelösten Skripte, die abgerufen werden sollen. Mögliche Werte sind:
    - all: Zeigt alle Details zur Jobausführung an, unabhängig davon, ob die Jobs noch laufen oder nicht.
    - running: zeigt nur Jobdetails für Jobs an, die gerade ausgeführt werden.
    - done: zeigt nur Jobdetails für Jobs an, deren Ausführung beendet ist.
    - killed: zeigt nur Jobdetails für Jobs an, deren Ausführung über den dedizierten Endpunkt gewaltsam beendet wurde (siehe unten).
    - triggered: zeigt nur Jobdetails für Jobs an, die ausgelöst, aber noch nicht gestartet wurden.
    - failed: zeigt nur Auftragsdetails für Jobs an, deren Ausführung als fehlgeschlagen markiert wurde.
    - \_namefilter (optional) \_: ruft nur Ausführungen für die angegebene Skript-ID ab.
  - Gibt eine Sammlung von Details zur Auftragsausführung als JSON zurück. Weitere Informationen finden Sie unter [Nachrichtenstruktur mit Details zur Jobausführung](#).

Beispielantwort:

```
[
  {
    "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"
  }
]
```

Details zur Auftragsausführung werden abgerufen

- Unterstützte Methode: GET

- Pfad: `/getjobexecutioninfo/{jobexecutionid:.+}`
- Argumente:
  - `jobexecutionid` (erforderlich): Die eindeutige Kennung zur Auftragsausführung, um die entsprechenden Details zur Jobausführung abzurufen.
- Gibt Folgendes zurück: eine JSON-Zeichenfolge, die einzelne Details zur Jobausführung darstellt (siehe [Nachrichtenstruktur mit Details zur Jobausführung](#)), oder eine leere Antwort, wenn für die angegebene Kennung keine Details zur Jobausführung gefunden werden konnten.

Listet asynchron gestartete Skripten auf, die beendet werden können

- Unterstützte Methode: GET
- Pfad: `/killablescripts`
- Gibt eine Sammlung von Bezeichnern für die Auftragsausführung von Jobs zurück, die asynchron gestartet wurden, derzeit noch ausgeführt werden und gewaltsam beendet werden können (siehe den `/kill` Endpunkt unten).

Listet synchron gestartete Skripten auf, die beendet werden können

- Unterstützte Methode: GET
- Pfad: `/killablesyncscripts`
- Gibt eine Sammlung von Identifikatoren für die Auftragsausführung von Jobs zurück, die synchron gestartet wurden, derzeit noch ausgeführt werden und gewaltsam beendet werden können (siehe den `/kill` Endpunkt unten).

Beenden einer bestimmten Jobausführung

- Unterstützte Methode: GET
- Pfad: `/kill/{identifizier:.+}`
- Argument: ID für die Auftragsausführung (erforderlich): Die eindeutige Kennung der Jobausführung, die auf die Jobausführung hinweist, die gewaltsam beendet werden soll.
- Gibt Folgendes zurück: eine Textnachricht, in der das Ergebnis des Abbruchs der Auftragsausführung detailliert beschrieben wird. Die Nachricht enthält die Skriptkennung, die eindeutige Kennung der Jobausführung sowie das Datum und die Uhrzeit, an dem die Ausführung

abgebrochen wurde. Wenn für den angegebenen Bezeichner keine laufende Jobausführung gefunden werden konnte, wird stattdessen eine Fehlermeldung zurückgegeben.

#### Warning

- Die Laufzeit bemüht sich nach besten Kräften, die Ausführung des Zieljobs ordnungsgemäß zu beenden. Daher kann es einige Zeit dauern, bis die Antwort vom /kill-Endpunkt den Aufrufer erreicht, da die AWS Blu-Age-Laufzeit versucht, die geschäftlichen Auswirkungen eines Abbruchs des Jobs zu minimieren.
- Das gewaltsame Abbrechen einer Auftragsausführung sollte nicht leichtfertig erfolgen, da dies direkte geschäftliche Folgen haben kann, einschließlich möglichen Datenverlusten oder Datenbeschädigungen. Sie sollte den Fällen vorbehalten bleiben, in denen die Ausführung eines bestimmten Auftrags schiefgegangen ist und die Mittel zur Datenbehebung eindeutig identifiziert wurden.
- Die Einstellung eines Arbeitsplatzes sollte zu weiteren Untersuchungen (Post-Mortem-Analyse) führen, um herauszufinden, was schief gelaufen ist, und um geeignete Abhilfemaßnahmen zu ergreifen.
- In jedem Fall wird der Versuch, einen laufenden Job zu beenden, in den Serverprotokollen mit Warnmeldungen protokolliert.

#### Auflisten vorhandener Checkpoints aus Gründen der Neustartfähigkeit

Die Neustartfähigkeit von Job hängt von der Fähigkeit der Skripte ab, Checkpoints zu registrieren, `CheckpointRegistry` um den Fortschritt der Auftragsausführung zu verfolgen. Wenn eine Auftragsausführung nicht ordnungsgemäß beendet wird und Neustart-Checkpoints registriert wurden, kann man die Jobausführung einfach vom letzten bekannten registrierten Checkpoint aus neu starten (ohne die Schritte über dem Checkpoint ausführen zu müssen).

- Unterstützte Methode: GET
- Pfad: `/restarts`
- Gibt die Liste der vorhandenen Neustartpunkte, die verwendet werden können, um einen Job neu zu starten, dessen Ausführung nicht ordnungsgemäß abgeschlossen wurde, als HTML-Seite zurück. Wenn keine Checkpoints von Skripten registriert wurden, lautet der Seiteninhalt „Keine registrierten Checkpoints“.

## Einen Job neu starten (synchron)

- Unterstützte Methode: GET
- Pfad: `/restart/{hashcode}`
- Argumente: Hashcode (Ganzzahl — verpflichtend): Startet eine zuvor abgebrochene Jobausführung neu und verwendet dabei den bereitgestellten Hashcode als Checkpoint-Wert (siehe `/restarts` Endpunkt oben, um zu erfahren, wie Sie einen gültigen Checkpoint-Wert abrufen können).
- Rücksendungen: siehe Beschreibung der Rücksendung oben. `script`

## Einen Job neu starten (asynchron)

- Unterstützte Methode: GET
- Pfad: `/triggerrestart/{hashcode}`
- Argumente: Hashcode (Ganzzahl — verpflichtend): Startet eine zuvor abgebrochene Jobausführung neu und verwendet dabei den bereitgestellten Hashcode als Checkpoint-Wert (siehe `/restarts` Endpunkt oben, um zu erfahren, wie Sie einen gültigen Checkpoint-Wert abrufen können).
- Rücksendungen: siehe Beschreibung der Rücksendung oben. `triggerscript`

## Einstellung des Thread-Limits für asynchrone Jobausführungen

Die asynchrone Ausführung des Jobs basiert auf einem dedizierten Thread-Pool in der JVM. Dieser Pool hat ein festes Limit in Bezug auf die Anzahl der verfügbaren Threads. Der Benutzer hat die Möglichkeit, das Limit an die Hostkapazitäten (Anzahl der CPUs, verfügbarer Speicher usw.) anzupassen. Standardmäßig ist das Thread-Limit auf 5 Threads festgelegt.

- Unterstützte Methode: GET
- Pfad: `/settriggerthreadlimit/{threadlimit:.+}`
- Argument (Ganzzahl): Das neue anzuwendende Thread-Limit. Muss eine strikt positive Ganzzahl sein.
- Gibt eine Nachricht (`String`) zurück, die das neue und das vorherige Thread-Limit angibt, oder eine Fehlermeldung, wenn der angegebene Thread-Grenzwert nicht gültig ist (keine strikt positive Ganzzahl).

**Beispielantwort:**

```
Set thread limit for Script Tower Control to 10 (previous value was 5)
```

**Zählung der aktuell ausgeführten, ausgelösten Jobausführungen**

- Unterstützte Methode: GET
- Pfad: /countrunningtriggeredscripts
- Gibt eine Meldung zurück, die die Anzahl der laufenden Jobs angibt, die asynchron gestartet wurden, und das Thread-Limit (das ist die maximale Anzahl von ausgelösten Jobs, die gleichzeitig ausgeführt werden können).

**Beispielantwort:**

```
0 triggered script(s) running (limit =10)
```

**Note**

Dies kann verwendet werden, um vor dem Starten eines Jobs zu überprüfen, ob das Thread-Limit nicht erreicht wurde (was verhindern würde, dass der Job gestartet werden könnte).

**Löscht die Informationen zur Auftragsausführung**

Die Informationen zur Auftragsausführung verbleiben im Serverspeicher, solange der Server aktiv ist. Es kann praktisch sein, die ältesten Informationen aus dem Speicher zu löschen, da sie nicht mehr relevant sind. Das ist der Zweck dieses Endpunkts.

- Unterstützte Methode: GET
- Pfad: /purgejobinformation/{age:.+}
- Argumente: Ein strikt positiver Ganzzahlwert, der das Alter der zu löschenden Informationen in Stunden angibt.
- Gibt eine Nachricht mit den folgenden Informationen zurück:
  - Name der Löschdatei, in der Informationen zur Ausführung gelöschter Jobs zu Archivierungszwecken gespeichert werden.
  - Anzahl der Informationen zur gelöschten Jobausführung.



- Anzahl der verbleibenden Informationen zur Jobausführung im Memo

## Metriken und Endpunkte

### JVM

Dieser Endpunkt gibt verfügbare Metriken zur JVM zurück.

- Unterstützte Methode: GET
- Pfad: `/metrics/jvm`
- Argumente: keine
- Gibt eine Nachricht mit den folgenden Informationen zurück:
  - `ThreadActiveCount`: Anzahl der aktiven Threads.
  - `jvmMemoryUsed`: Speicher, der von der Java Virtual Machine aktiv genutzt wird.
  - `jvmMemoryMax`: Maximal zulässiger Arbeitsspeicher für die Java Virtual Machine.
  - `jvmMemoryFree`: Verfügbarer Speicher, der derzeit nicht von der Java Virtual Machine verwendet wird.

### Sitzung

Dieser Endpunkt gibt Metriken zurück, die sich auf aktuell geöffnete HTTP-Sitzungen beziehen.

- Unterstützte Methode: GET
- Pfad: `/metrics/session`
- Argumente: keine
- Gibt eine Nachricht mit den folgenden Informationen zurück:
  - `sessionCount`: Anzahl der aktiven Benutzersitzungen, die derzeit vom Server verwaltet werden.

### Stapel

- Unterstützte Methode: GET
- Pfad: `/metrics/batch`
- Argumente:
  - `startTimeStamp` (optional, Zahl): Startzeitstempel für die Datenfilterung.

- endTimeStamp (optional, Zahl): Endzeitstempel für die Datenfilterung.
- Seite (optional, Zahl): Seitennummer für die Paginierung.
- pageSize (optional, Anzahl): Anzahl der Elemente pro Seite bei der Paginierung.
- Gibt eine Nachricht mit den folgenden Informationen zurück:
  - Inhalt: Liste der Metriken zur Batch-Ausführung.
  - pageNumber: Aktuelle Seitennummer bei der Paginierung.
  - pageSize: Anzahl der pro Seite angezeigten Elemente.
  - totalPages: Gesamtzahl der verfügbaren Seiten.
  - AnzahlOfElements: Anzahl der Elemente auf der aktuellen Seite.
  - last: Boolesches Kennzeichen für die letzte Seite.
  - first: Boolesche Flagge für die erste Seite.

## Transaktion

- Unterstützte Methode: GET
- Pfad: /metrics/transaction
- Argumente:
  - startTimeStamp (optional, Zahl): Startzeitstempel für die Datenfilterung.
  - endTimeStamp (optional, Zahl): Endzeitstempel für die Datenfilterung.
  - Seite (optional, Zahl): Seitennummer für die Paginierung.
  - pageSize (optional, Anzahl): Anzahl der Elemente pro Seite bei der Paginierung.
- Gibt eine Nachricht mit den folgenden Informationen zurück:
  - Inhalt: Liste der Metriken zur Transaktionsausführung.
  - pageNumber: Aktuelle Seitennummer bei der Paginierung.
  - pageSize: Anzahl der pro Seite angezeigten Elemente.
  - totalPages: Gesamtzahl der verfügbaren Seiten.
  - AnzahlOfElements: Anzahl der Elemente auf der aktuellen Seite.
  - last: Boolesches Kennzeichen für die letzte Seite.
  - first: Boolesche Flagge für die erste Seite.

## Andere Endpunkte

Verwenden Sie diese Endpunkte, um registrierte Programme oder Dienste aufzulisten, den Systemstatus zu ermitteln und JICS-Transaktionen zu verwalten.

### Themen

- [Registrierte Programme auflisten](#)
- [Auflisten registrierter Dienste](#)
- [Gesundheitsstatus](#)
- [Listet die verfügbaren JICS-Transaktionen auf](#)
- [Starten Sie eine JICS-Transaktion](#)
- [Starten Sie eine JICS-Transaktion \(Alternative\)](#)

### Registrierte Programme auflisten

- Unterstützte Methode: GET
- Pfad: `/programs`
- Gibt die Liste der registrierten Programme als HTML-Seite zurück. Jedes Programm wird durch seine Hauptprogramm-ID bezeichnet. Sowohl modernisierte Legacy-Programme als auch Hilfsprogramme (IDCAMS, IEBGENER, etc...) werden wieder in die Liste aufgenommen. Bitte beachten Sie, dass die verfügbaren Hilfsprogramme von den Utility-Webanwendungen abhängen, die auf Ihrem Tomcat-Server bereitgestellt wurden. Beispielsweise sind die z/OS-Utility-Supportprogramme für modernisierte iSeries-Geräte möglicherweise nicht verfügbar, da sie nicht relevant sind.

### Auflisten registrierter Dienste

- Unterstützte Methode: GET
- Pfad: `/services`
- Gibt die Liste der registrierten Runtime-Dienste als HTML-Seite zurück. Die angegebenen Dienste werden von der AWS Blu Age-Laufzeit als Hilfsprogramme bereitgestellt, die zum Beispiel in Groovy-Skripten verwendet werden können. Blusam-Ladedienste (um Blusam-Datensätze aus älteren Datensätzen zu erstellen) fallen in diese Kategorie.

### Beispielantwort:

```
<p>BluesamESDSFileLoader</p><p>BluesamKSDSFileLoader</p><p>BluesamRRDSFileLoader</p>
```

## Gesundheitsstatus

- Unterstützte Methode: GET
- Pfad: /
- Gibt eine einfache Nachricht zurück, die anzeigt, dass die Gapwalk-Anwendung läuft () `Jics application is running.`

## Listet die verfügbaren JICS-Transaktionen auf

- Unterstützte Methode: GET
- Pfad: /transactions
- Gibt eine HTML-Seite zurück, auf der alle verfügbaren JICS-Transaktionen aufgelistet sind. Dies ist nur für Umgebungen mit JICS-Elementen (Modernisierung älterer CICS-Elemente) sinnvoll.

## Beispielantwort:

```
<p>INQ1</p><p>MENU</p><p>MNT2</p><p>ORD1</p><p>PRNT</p>
```

## Starten Sie eine JICS-Transaktion

- Unterstützte Methoden: GET, POST
- Pfad: /jicstransrunner/{jtrans:.+}
- Argumente:
  - JICS-Transaktions-ID (Zeichenfolge, erforderlich): Kennung der JICS-Transaktion, die gestartet werden soll (max. 8 Zeichen lang)
  - <String, Object>erforderlich: zusätzliche Eingabedaten, die an die Transaktion übergeben werden sollen, als Map. Der Inhalt dieser Karte wird verwendet, um die [COMMAREA](#) mit Strom zu versorgen, die im Rahmen der JICS-Transaktion verbraucht wird. Die Map kann leer sein, wenn für die Ausführung der Transaktion keine Daten erforderlich sind.
  - optional: HTTP-Header-Einträge, um die Ausführungsumgebung für die angegebene Transaktion anzupassen. Die folgenden Header-Schlüssel werden unterstützt:
    - `jics-channel`: Der Name des JICS-KANALS, der von dem Programm verwendet werden soll, das bei diesem Transaktionsstart gestartet wird.

- `jics-container`: Der Name des JICS-CONTAINERS, der für diesen JICS-Transaktionsstart verwendet werden soll.
  - `jics-startcode`: der STARTCODE (Zeichenfolge, bis zu 2 Zeichen), der beim Start der JICS-Transaktion verwendet werden soll. Mögliche Werte finden Sie unter [STARTCODE](#) (auf der Seite nach unten blättern).
  - `jicxa-xid`: Die XID (X/Open Transaction Identifier XID-Struktur) einer „globalen Transaktion“ ([XA](#)), die vom Aufrufer initiiert wurde und an der der aktuelle Start der JICS-Transaktion beteiligt sein wird.
- Gibt zurück: eine `com.netfective.bluage.gapwalk.rt.shared.web.TransactionResultBean` JSON-Serialisierung, die das Ergebnis des Starts der JICS-Transaktion darstellt.

Weitere Hinweise zu den Einzelheiten der Struktur finden Sie unter. [Struktur der Ergebnisse des Transaktionsstarts](#)

Starten Sie eine JICS-Transaktion (Alternative)

- unterstützte Methoden: GET, POST
- Pfad: `/jicstransaction/{jtrans: .+}`
- Argumente:  
JICS-Transaktions-ID (Zeichenfolge, erforderlich)

Kennung der JICS-Transaktion, die gestartet werden soll (max. 8 Zeichen lang)

erforderlich: zusätzliche Eingabedaten, die an die Transaktion übergeben werden sollen, als `Map<String, Object>`

Der Inhalt dieser Karte wird verwendet, um die [COMMAREA](#) mit Strom zu versorgen, die im Rahmen der JICS-Transaktion verbraucht wird. Die Map kann leer sein, wenn für die Ausführung der Transaktion keine Daten erforderlich sind.

optional: HTTP-Header-Einträge, um die Ausführungsumgebung für die angegebene Transaktion anzupassen.

Die folgenden Header-Schlüssel werden unterstützt:

- `jics-channel`: Der Name des JICS-KANALS, der von dem Programm verwendet werden soll, das bei diesem Transaktionsstart gestartet wird.

- `jics-container`: Der Name des JICS-CONTAINERS, der für diesen JICS-Transaktionsstart verwendet werden soll.
  - `jics-startcode`: der STARTCODE (Zeichenfolge, bis zu 2 Zeichen), der beim Start der JICS-Transaktion verwendet werden soll. Mögliche Werte finden Sie unter [STARTCODE](#) (auf der Seite nach unten blättern).
  - `jicxa-xid`: Die XID (X/Open Transaction Identifier XID-Struktur) einer „globalen Transaktion“ ([XA](#)), die vom Aufrufer initiiert wurde und an der der aktuelle Start der JICS-Transaktion beteiligt sein wird.
- gibt zurück: eine `com.netfective.bluage.gapwalk.rt.shared.web.RecordHolderBean` JSON-Serialisierung, die das Ergebnis des Starts der JICS-Transaktion darstellt. Die Einzelheiten der Struktur finden Sie in [Aufbau des Transaktionsstarts, Aufzeichnung der Ergebnisse](#)

## Endpunkte im Zusammenhang mit Job Queues

Job Queues sind die AWS Blu-Age-Unterstützung für den AS400-Mechanismus zur Einreichung von Jobs. Job-Warteschlangen werden in AS400 verwendet, um Jobs in bestimmten Thread-Pools auszuführen. Eine Job-Warteschlange wird durch einen Namen und eine maximale Anzahl von Threads definiert, die der maximalen Anzahl von Programmen entspricht, die gleichzeitig in dieser Warteschlange ausgeführt werden können. Wenn mehr Jobs in der Warteschlange eingereicht werden als die maximale Anzahl von Threads, warten Jobs darauf, dass ein Thread verfügbar ist.

Eine vollständige Statusliste für einen Job in einer Warteschlange finden Sie unter [Möglicher Status eines Jobs in einer Warteschlange](#).

Vorgänge in Auftragswarteschlangen werden über die folgenden dedizierten Endpunkte abgewickelt. Sie können diese Operationen über die URL der Gapwalk-Anwendung mit der folgenden Stamm-URL aufrufen: `http://server:port/gapwalk-application/jobqueue`

### Themen

- [Verfügbare Warteschlangen auflisten](#)
- [Starten Sie eine Job-Warteschlange oder starten Sie sie neu](#)
- [Reichen Sie einen Job zum Start ein](#)
- [Listet alle eingereichten Jobs auf](#)
- [Geben Sie alle Jobs frei, die sich in der Warteschleife befinden](#)
- [Geben Sie alle Jobs frei, die für einen bestimmten Jobnamen „in der Warteschleife“ sind](#)

- [Gibt einen bestimmten Job für eine Jobnummer frei](#)
- [Reichen Sie einen Job nach einem sich wiederholenden Zeitplan ein](#)
- [Listet alle eingereichten, sich wiederholenden Jobs auf](#)
- [Brechen Sie die Planung eines sich wiederholenden Jobs ab](#)

### Verfügbare Warteschlangen auflisten

- Unterstützte Methode: GET
- Pfad: `list-queues`
- Gibt die Liste der verfügbaren Warteschlangen zusammen mit ihrem Status als JSON-Liste von Schlüsselwerten zurück.

### Beispielantwort:

```
{"Default":"STAND_BY", "queue1":"STARTED", "queue2":"STARTED"}
```

### Mögliche Status für eine Job-Warteschlange sind:

#### STAND\_BY

Die Job-Warteschlange wartet darauf, gestartet zu werden.

#### STARTED

Die Job-Warteschlange ist aktiv und läuft.

#### UNKNOWN

Der Status der Auftragswarteschlange kann nicht ermittelt werden.

### Starten Sie eine Job-Warteschlange oder starten Sie sie neu

- Unterstützte Methode: POST
- Pfad: `/restart/{name}`
- Argument: Der Name der Warteschlange, die gestartet/neu gestartet werden soll, als Zeichenfolge - obligatorisch.
- Der Endpunkt gibt nichts zurück, sondern stützt sich auf den HTTP-Status, um das Ergebnis des Start-/Neustartvorgangs anzuzeigen:

## HTTP 200

Der Start-/Neustartvorgang verlief gut: Die angegebene Jobwarteschlange ist jetzt GESTARTET.

## HTTP 404

Die Jobwarteschlange existiert nicht.

## HTTP 503

Beim Start-/Neustartversuch ist eine Ausnahme aufgetreten (Serverprotokolle sollten überprüft werden, um herauszufinden, was schief gelaufen ist).

Reichen Sie einen Job zum Start ein

- Unterstützte Methode: POST
- Pfad: `/submit`
- Argument: verpflichtend als Hauptteil der Anfrage, eine JSON-Serialisierung eines `com.netfective.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` Objekts. Weitere Informationen finden Sie unter [Job abschicken und Job-Eingabe planen](#).
- Gibt zurück: ein JSON, das das Original enthält, `SubmitJobMessage` und ein Protokoll, das angibt, ob der Job eingereicht wurde oder nicht.

Listet alle eingereichten Jobs auf

- Unterstützte Methode: GET
- Pfad: `/list-jobs?status={status}&size={size}&page={page}&sort={sort}`
- Argumente:
  - Seite: Seitennummer, die abgerufen werden soll (Standard = 1)
  - Größe: Größe der Seite (Standard = 50, max = 300)
  - sort: Die Reihenfolge der Jobs. (Standard = „ExecutionID“). „ExecutionId“ ist derzeit der einzige unterstützte Wert
  - status: (optional) Falls vorhanden, wird nach dem Status gefiltert.
- Gibt Folgendes zurück: eine Liste aller geplanten Jobs als JSON-Zeichenfolge. Eine Beispielantwort finden Sie unter [Liste der Antworten auf geplante Jobs](#).



Geben Sie alle Jobs frei, die sich in der Warteschleife befinden

- Unterstützte Methode: POST
- Pfad: `/release-all`
- Rückgabe: eine Meldung, die das Ergebnis des Release-Versuchs angibt. Zwei mögliche Fälle sind hier:
  - HTTP 200 und eine Meldung „Alle Jobs wurden erfolgreich veröffentlicht!“ wenn alle Jobs erfolgreich veröffentlicht wurden.
  - HTTP 503 und eine Meldung „Jobs nicht veröffentlicht. Ein unbekannter Fehler ist aufgetreten. Weitere Informationen finden Sie im Protokoll“, falls beim Release-Versuch ein Fehler aufgetreten ist.

Geben Sie alle Jobs frei, die für einen bestimmten Jobnamen „in der Warteschleife“ sind

`<job name, job number>`Für einen bestimmten Jobnamen können mehrere Jobs mit unterschiedlichen Job-Nummern eingereicht werden (die Einheitlichkeit eines ausgeführten Jobs wird durch mehrere garantiert). Der Endpunkt versucht, alle Jobübermittlungen mit dem angegebenen Jobnamen, die sich in der Warteschleife befinden, freizugeben.

- Unterstützte Methode: POST
- Pfad: `/release/{name}`
- Argumente: Der Jobname, nach dem gesucht werden soll, als Zeichenfolge. Zwingend erforderlich.
- Rückgabe: eine Meldung, die das Ergebnis des Freigabeversuchs angibt. Zwei mögliche Fälle sind hier:
  - HTTP 200 und eine Meldung „Jobs in Group `<name>`(`<number of released jobs>`) erfolgreich veröffentlicht!“ Jobs wurden erfolgreich veröffentlicht.
  - HTTP 503 und die Meldung „Jobs in der Gruppe `<name>`wurden nicht veröffentlicht. Ein unbekannter Fehler ist aufgetreten. Weitere Informationen finden Sie im Protokoll“, falls beim Release-Versuch ein Fehler aufgetreten ist.

Gibt einen bestimmten Job für eine Jobnummer frei

Der Endpunkt versucht, die eindeutige Jobübermittlung freizugeben, die für das angegebene Paar „in der Warteschleife“ ist `<job name, job number>`.

- Unterstützte Methode: POST

- Pfad: `/release/{name}/{number}`

- Argumente:

Name

der Jobname, nach dem gesucht werden soll, als Zeichenfolge. Zwingend erforderlich.

Zahl

die Jobnummer, nach der gesucht werden soll, als Ganzzahl. Zwingend erforderlich.

returns

eine Meldung, die das Ergebnis des Freigabeversuchs angibt. Zwei mögliche Fälle sind hier:

- HTTP 200 und die Meldung „Job <name/number> erfolgreich veröffentlicht!“ wenn der Job erfolgreich veröffentlicht wurde.
- HTTP 503 und die Meldung „Job <name/number> nicht veröffentlicht. Ein unbekannter Fehler ist aufgetreten. Weitere Informationen finden Sie im Protokoll“, falls beim Release-Versuch ein Fehler aufgetreten ist.

Reichen Sie einen Job nach einem sich wiederholenden Zeitplan ein

Planen Sie einen Job, der mit einem sich wiederholenden Zeitplan ausgeführt wird.

- Unterstützte Methode: POST
- Pfad: `/schedule`
- Argument: Der Anfragetext muss eine JSON-Serialisierung eines `com.netfactive.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` Objekts enthalten.

Listet alle eingereichten, sich wiederholenden Jobs auf

- Unterstützte Methode: GET
- Pfad: `/schedule/list?status={status}&size={size}&page={page}&sort={sort}`
- Argumente:
  1. Seite: Seitennummer, die abgerufen werden soll (Standard = 1)
  2. Größe: Größe der Seite (Standard = 50, max = 300)
  3. sort: Die Reihenfolge der Jobs. (Standard = „id“). „id“ ist derzeit der einzige unterstützte Wert.

4. status: (optional) Falls vorhanden, wird nach dem Status gefiltert. Mögliche Werte sind die in Abschnitt 1 genannten.
5. status: (optional) Falls vorhanden, wird nach dem Status gefiltert. Mögliche Werte sind die in Abschnitt 1 genannten.
6. Gibt Folgendes zurück: eine Liste aller geplanten Jobs als JSON-Zeichenfolge.

Brechen Sie die Planung eines sich wiederholenden Jobs ab

Entfernt einen Job, der nach einem sich wiederholenden Zeitplan erstellt wurde. Der Status der Auftragsplanung ist auf INAKTIV gesetzt.

- Unterstützte Methode: GET
- Pfad: `/schedule/remove/{schedule_id}`
- Argument: `schedule_id`, der Bezeichner des geplanten Jobs, der entfernt werden soll.

## REST-Endpunkte der Blusam Application Console

Die Blusam Application Console ist eine API, die entwickelt wurde, um die Verwaltung modernisierter VSAM-Datensätze zu vereinfachen. Endpunkte für die Blusam-Webanwendung verwenden den Root-Pfad. `/bac`

Themen

- [Datensätze und zugehörige Endpunkte](#)
- [Massendatensätze, verwandte Endpunkte](#)
- [Datensätze](#)
- [Masks](#)
- [Sonstige](#)
- [Benutzer](#)

## Datensätze und zugehörige Endpunkte

Verwenden Sie die folgenden Endpunkte, um einen bestimmten Datensatz zu erstellen oder zu verwalten.

Themen

- [Erstellen Sie einen Datensatz](#)
- [Hochladen einer Datei](#)
- [Lädt einen Datensatz \(POST\)](#)
- [Lädt einen Datensatz \(GET\)](#)
- [Einen Datensatz aus einem Amazon S3 S3-Bucket laden](#)
- [Exportieren Sie einen Datensatz in einen Amazon S3 S3-Bucket](#)
- [Einen Datensatz löschen](#)
- [Löscht einen Datensatz](#)
- [Zählen Sie Datensätze](#)

## Erstellen Sie einen Datensatz

Sie können diesen Endpunkt verwenden, um eine Datensatzdefinition zu erstellen.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/createDataSet
- Argumente:

Name

(erforderlich, Zeichenfolge): der Name des Datensatzes.

Typ

(erforderlich, Zeichenfolge): der Datensatztyp. Mögliche Werte sind:ESDS,KSDS,RRDS.

Datensatzgröße

(optional, Zeichenfolge): Maximale Größe jedes Datensatzes des Datensatzes.

Feste Länge

(optional, boolean): Gibt an, ob die Länge des Datensatzes fest ist.

Kompression

(optional, boolean): Gibt an, ob der Datensatz komprimiert ist.

CacheEnable

(optional, boolean): Gibt an, ob das Caching für den Datensatz aktiviert ist.

## Alternative Schlüssel

(optional, Liste der Schlüssel):

- Offset (erforderlich, Zahl)
  - Länge (erforderlich, Zahl)
  - Name (erforderlich, Nummer)
- Gibt eine JSON-Datei zurück, die den neu erstellten Datensatz darstellt.

### Beispielanfrage:

```
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
}
```

### Beispielantwort:

```
{
  "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
}
```

## Hochladen einer Datei

Sie können diesen Endpunkt verwenden, um Dateien auf den Server hochzuladen. Die Datei wird in einem temporären Ordner gespeichert, der jedem bestimmten Benutzer entspricht. Verwenden Sie diesen Endpunkt jedes Mal, wenn Sie eine Datei hochladen müssen.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/upload
- Argumente:
  - file
    - (erforderlich, multipart/form-data): Die hochzuladende Datei.
- Gibt einen booleschen Wert zurück, der den Status des Uploads widerspiegelt

### Lädt einen Datensatz (POST)

Nachdem Sie `createDataSet` die Datensatzdefinition erstellt haben, können Sie Datensätze, die der hochgeladenen Datei zugeordnet sind, in einen bestimmten Datensatz laden.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/loadDataSet
- Argumente:
  - Name
    - (erforderlich, Zeichenfolge): der Name des Datensatzes.
- Gibt den Status der Anfrage und des geladenen Datensatzes zurück.

### Lädt einen Datensatz (GET)

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/loadDataSet
- Argumente:
  - Name
    - (erforderlich, Zeichenfolge): der Name des Datensatzes.
  - Datensatz-Datei
    - (erforderlich, Zeichenfolge): Der Name der Datensatzdatei.

- Gibt den Status der Anfrage und des geladenen Datensatzes zurück.

## Einen Datensatz aus einem Amazon S3 S3-Bucket laden

Lädt einen Datensatz mithilfe einer Listcat-Datei aus einem Amazon S3 S3-Bucket.

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: `/api/services/rest/bluesamservice/loadDataSetFromS3`
- Argumente:

Speicherort von CatFiles3 auflisten

(erforderlich, Zeichenfolge): Der Amazon S3 S3-Speicherort der Listcat-Datei.

DatasetFiles3-Speicherort

(erforderlich, Zeichenfolge): Der Amazon S3 S3-Speicherort der Datensatzdatei.

Region

(erforderlich, Zeichenfolge): Amazon S3, AWS-Region in dem die Dateien gespeichert sind.

- Gibt den neu erstellten Datensatz zurück

Beispielanfrage:

```
/BAC/api/services/rest/bluesamservice/loadDataSetFromS3?region=us-east-1&listcatFileS3Location=s3://bucket-name/listcat.json&datasetFileS3Location=s3://bucket-name/dataset.DAT
```

## Exportieren Sie einen Datensatz in einen Amazon S3 S3-Bucket

Exportiert einen Datensatz in den angegebenen Amazon S3 S3-Bucket.

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: `/api/services/rest/bluesamservice/exportDataSetToS3`
- Argumente:



## s3Location

(erforderlich, Zeichenfolge): Der Amazon S3 S3-Speicherort, an den der Datensatz exportiert werden soll.

## datasetName

(erforderlich, Zeichenfolge): Der Name des Datensatzes, der exportiert werden soll.

## Region

(erforderlich, Zeichenfolge): der AWS-Region des Amazon S3 S3-Buckets.

- Gibt den exportierten Datensatz zurück

### Beispielanfrage:

```
/BAC/api/services/rest/bluesamservice/exportDataSetToS3?region=eu-west-1&s3Location=s3://bucket-name/dump&datasetName=dataset
```

## Einen Datensatz löschen

Löscht alle Datensätze aus einem Datensatz.

- Unterstützte Methoden: POST, GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/clearDataSet
- Argumente:

### Name

(erforderlich, Zeichenfolge): Der Name des zu löschenden Datensatzes.

- Gibt den Status der Anfrage zurück.

## Löscht einen Datensatz

Löscht die Datensatzdefinition und die Datensätze.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/deleteDataSet

- **Argumente:**

Name

(erforderlich, Zeichenfolge): Der Name des zu löschenden Datensatzes.

- Gibt den Status der Anfrage und den gelöschten Datensatz zurück.

## Zählen Sie Datensätze

Dieser Endpunkt gibt die Anzahl der Datensätze zurück, die einem Datensatz zugeordnet sind.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/countRecords
- Argumente:

Name

(erforderlich, Zeichenfolge): der Name des Datensatzes.

- Gibt zurück: die Anzahl der Datensätze

## Massendatensätze, verwandte Endpunkte

Verwenden Sie die folgenden Endpunkte, um mehrere Datensätze gleichzeitig zu erstellen oder zu verwalten.

### Themen

- [Datensätze exportieren \(GET\)](#)
- [Datensätze exportieren \(POST\)](#)
- [Erstellen Sie mehrere Datensätze](#)
- [Listet alle Datensätze auf](#)
- [Direkte Liste aller Datensätze](#)
- [Direkte Auflistung aller Datensätze pro Seite](#)
- [Datensatz streamen](#)
- [Löscht alle Datensätze](#)
- [Ruft Datensatzdefinitionen aus der Listcat-Datei ab](#)

- [Ruft Datensatzdefinitionen aus der hochgeladenen List-Cat-Datei ab](#)
- [Holen Sie sich einen Datensatz](#)
- [Listcat aus der JSON-Datei laden](#)

### Datensätze exportieren (GET)

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/exportDataSet
- Argumente:

datasetName

(erforderlich, Zeichenfolge): Der Name des zu exportierenden Datensatzes.

Datensatz OutputFile

(erforderlich, Zeichenfolge): Der Pfad des Ordners, in dem Sie den exportierten Datensatz auf dem Server speichern möchten.

rdw

(erforderlich, boolean): ob das Datensatzdeskriptorwort (RDW) Teil der exportierten Datensätze sein soll. Wenn der Datensatz Datensätze mit fester Länge enthält, wird der Wert dieses Parameters ignoriert.

- Gibt den Status der Anfrage und den Pfad zu der Datei zurück, die den exportierten Datensatz enthält (falls vorhanden). Wenn der Datensatz in der Antwort Null ist, bedeutet das, dass das System keinen Datensatz mit dem angegebenen Namen finden konnte.

### Datensätze exportieren (POST)

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/exportDataSet
- Argumente:

Parameter auswerfen

(erforderlich, BACReadParameters): Bluesam-Leseparameter.

- Gibt den Status des exportierten Datensatzes zurück.

Erstellen Sie mehrere Datensätze

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: `/api/services/rest/bluesamservice/createAllDataSets`
- Argumente:
  - Liste der Datensätze

Name

(erforderlich, Zeichenfolge): der Name des Datensatzes.

Typ

(erforderlich, Zeichenfolge): der Datensatztyp. Mögliche Werte sind:ESDS,KSDS,RRDS.

Datensatzgröße

(optional, Zeichenfolge): Maximale Größe jedes Datensatzes des Datensatzes.

Feste Länge

(optional, boolean): Gibt an, ob die Länge des Datensatzes fest ist.

Kompression

(optional, boolean): Gibt an, ob der Datensatz komprimiert ist.

CacheEnable

(optional, boolean): Gibt an, ob das Caching für den Datensatz aktiviert ist.

- Gibt Folgendes zurück: den Status der Anfrage und den neu erstellten Datensatz.

Listet alle Datensätze auf

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/bluesamservice/listDataSet`

- Argumente: Keine

- Gibt den Status der Anfrage und die Liste der Datensätze zurück.

### Direkte Liste aller Datensätze

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/bluesamservice/directListDataSet`
- Argumente: Keine
- Gibt den Status der Anfrage und die Liste der Datensätze zurück.

### Direkte Auflistung aller Datensätze pro Seite

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/bluesamservice/directListDataSetByPage`
- Argumente:

`datasetName`

(erforderlich, Zeichenfolge): der Name des Datensatzes.

`Seitennummer`

(erforderlich, int): Die Seitennummer.

`pageSize`

(erforderlich, int): die Seitengröße.

- Gibt den Status der Anfrage und die Liste der Datensätze zurück.

### Datensatz streamen

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: `/api/services/rest/bluesamservice/streamDataset`
- Argumente:

## datasetName

(erforderlich, Zeichenfolge): der Name des Datensatzes.

- Gibt zurück: Ein Stream der angeforderten Datensätze.

## Löscht alle Datensätze

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/removeAll
- Argumente: Keine
- Gibt zurück: einen booleschen Wert, der den Status der Anfrage darstellt.

## Ruft Datensatzdefinitionen aus der Listcat-Datei ab

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/getDataSetsDefinitionFromListcat
- Argumente:

Parameter FilePath

(erforderlich, Zeichenfolge): Der Pfad zur Listcat-Datei.

- Gibt zurück: eine Liste von Datensätzen

## Ruft Datensatzdefinitionen aus der hochgeladenen List-Cat-Datei ab

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/getDataSetsDefinitionFromUploadedListcat
- Argumente: Keine
- Gibt zurück: eine Liste von Datensätzen

## Holen Sie sich einen Datensatz

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/getDataSet
- Argumente:
  - Name
    - (erforderlich, Zeichenfolge): der Name des Datensatzes.
- Gibt den angeforderten Datensatz zurück.

## Listcat aus der JSON-Datei laden

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/loadListcatFromJsonFile
- Argumente:
  - filePath
    - (erforderlich, Zeichenfolge): Der Pfad zur Listcat-Datei.
- Gibt zurück: eine Liste von Datensätzen

## Datensätze

Verwenden Sie die folgenden Endpunkte, um Datensätze innerhalb eines Datensatzes zu erstellen oder zu verwalten.

### Themen

- [Erstellen eines Datensatzes](#)
- [Liest einen Datensatz](#)
- [Einen Datensatz löschen](#)
- [Einen Datensatz aktualisieren](#)
- [Speichert einen Datensatz](#)
- [Validiert einen Datensatz](#)

- [Holen Sie sich einen Datensatzbaum](#)

## Erstellen eines Datensatzes

Sie können diesen Endpunkt verwenden, um einen neuen Datensatz zu erstellen.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/createRecord`
- Argumente:

`dataset`

(erforderlich, DataSet): das Datensatz-Objekt

`maskieren`

(erforderlich, Maske): das Maskenobjekt.

- Gibt den Status der Anfrage und des erstellten Datensatzes zurück.

## Liest einen Datensatz

Sie können diesen Endpunkt verwenden, um einen Datensatz zu lesen.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/readDataSet`
- Argumente:

`dataset`

(erforderlich, DataSet): das Datensatz-Objekt.

- Gibt den Status der Anfrage und den Datensatz mit den Datensätzen zurück.

## Einen Datensatz löschen

Sie können diesen Endpunkt verwenden, um einen Datensatz aus einem Datensatz zu löschen.

- Unterstützte Methoden: POST



- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/crud/deleteRecord
- Argumente:

dataset

(erforderlich, DataSet): das Datensatz-Objekt

record

(erforderlich, Datensatz): der zu löschende Datensatz

- Gibt anschließend den Status der Löschung zurück.

### Einen Datensatz aktualisieren

Sie können diesen Endpunkt verwenden, um einen Datensatz zu aktualisieren, der einem Datensatz zugeordnet ist.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/crud/updateRecord
- Argumente:

dataset

(erforderlich, DataSet): das Datensatz-Objekt

record

(erforderlich, Datensatz): der zu aktualisierende Datensatz

- Gibt den Status der Anfrage und den Datensatz mit den Datensätzen zurück.

### Speichert einen Datensatz

Sie können diesen Endpunkt verwenden, um einen Datensatz in einem Datensatz zu speichern und dabei eine Maske zu verwenden.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/crud/saveRecord

- Argumente:

dataset

(erforderlich, DataSet): das Datensatz-Objekt

record

(erforderlich, Datensatz): der zu speichernde Datensatz

- Gibt den Status der Anfrage und den Datensatz mit den Datensätzen zurück.

### Validiert einen Datensatz

Verwenden Sie diesen Endpunkt, um einen Datensatz zu validieren.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/validateRecord`
- Argumente:  
dataset

(erforderlich, DataSet): das Datensatz-Objekt

- Gibt den Status der Anfrage und den Datensatz mit den Datensätzen zurück.

### Holen Sie sich einen Datensatzbaum

Verwenden Sie diesen Endpunkt, um den hierarchischen Baum eines Datensatzes abzurufen.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/getRecordTree`
- Argumente:  
dataset

(erforderlich, DataSet): das Datensatz-Objekt

record

(erforderlich, Datensatz): der Datensatz, der abgerufen werden soll

- Gibt den Status der Anfrage und den hierarchischen Baum des angeforderten Datensatzes zurück.

## Masks

Verwenden Sie die folgenden Endpunkte, um Masken zu laden oder auf einen Datensatz anzuwenden.

### Themen

- [Masken laden](#)
- [Maske auftragen](#)
- [Maskenfilter anwenden](#)

### Masken laden

Sie können diesen Endpunkt verwenden, um alle Masken abzurufen, die einem bestimmten Datensatz zugeordnet sind.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/loadMasks`
- Argumente:  
dataset

(erforderlich, DataSet): das Datensatz-Objekt

- Gibt den Status der Anfrage und die Liste der Masken zurück.

### Maske auftragen

Sie können diesen Endpunkt verwenden, um eine Maske auf einen bestimmten Datensatz anzuwenden.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/applyMask`
- Argumente:

## dataset

(erforderlich, DataSet): das Datensatz-Objekt

## maskieren

(erforderlich, Maske): das Datensatz-Objekt

- Gibt den Status der Anfrage und den Datensatz mit der angewendeten Maske zurück.

## Maskenfilter anwenden

Sie können diesen Endpunkt verwenden, um eine Maske und einen Filter auf einen bestimmten Datensatz anzuwenden.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: `/api/services/rest/crud/applyMaskFilter`
- Argumente:

## dataset

(erforderlich, DataSet): das Datensatz-Objekt

## maskieren

(erforderlich, Maske): das Datensatz-Objekt

- Gibt den Status der Anfrage und den Datensatz mit der angewendeten Maske und dem Filter zurück.

## Sonstige

Verwenden Sie die folgenden Endpunkte, um den Cache für einen Datensatz zu verwalten oder die Eigenschaften eines Datensatzes zu überprüfen

## Themen

- [Überprüfen Sie den Aufwärm-Cache](#)
- [Prüfen Sie, ob der Cache aktiviert](#)
- [Cache aktivieren](#)
- [Überprüfen Sie den zugewiesenen RAM-Cache](#)

- [Überprüfen Sie die Persistenz](#)
- [Überprüfen Sie die unterstützten Datensatztypen](#)
- [Überprüfen Sie den Serverstatus](#)

## Überprüfen Sie den Aufwärm-Cache

Prüft, ob der Aufwärm-Cache für einen bestimmten Datensatz aktiviert ist.

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/warmupCache
- Argumente:

Name

(erforderlich, Zeichenfolge): der Name des Datensatzes.

- Gibt Folgendes zurück: true, wenn der Warmup-Cache aktiviert ist, andernfalls false.

## Prüfen Sie, ob der Cache aktiviert

Prüft, ob der Cache für einen bestimmten Datensatz aktiviert ist.

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/isEnableCache
- Argumente: Keine
- Gibt true zurück, wenn das Caching aktiviert ist.

## Cache aktivieren

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rollen ROLE\_ADMIN und ROLE\_SUPER\_ADMIN.
- Pfad: /api/services/rest/bluesamservice/enableDisableCache/{enable}
- Argumente:

## enable

(erforderlich, boolean): Wenn auf true gesetzt, wird das Caching aktiviert.

- Gibt None zurück

## Überprüfen Sie den zugewiesenen RAM-Cache

Sie können diesen Endpunkt verwenden, um den zugewiesenen RAM-Cache-Speicher abzurufen.

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/allocatedRamCache
- Argumente: Keine
- Gibt zurück: die Größe des Speichers als Zeichenfolge

## Überprüfen Sie die Persistenz

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Pfad: /api/services/rest/bluesamservice/persistence
- Argumente: Keine
- Gibt zurück: die als Zeichenfolge verwendete Persistenz

## Überprüfen Sie die unterstützten Datensatztypen

- Unterstützte Methoden: GET
- Pfad: /api/services/rest/bluesamservice/getDataSetTypes
- Erfordert Authentifizierung und die Rolle ROLE\_USER.
- Argumente: Keine
- Gibt Folgendes zurück: die Liste der unterstützten Datensatztypen als Liste von Zeichenketten.

## Überprüfen Sie den Serverstatus

- Unterstützte Methoden: GET

- Pfad: `/api/services/rest/bluesamserver/serverIsUp`
- Argumente: Keine
- Gibt zurück: Keine. Der HTTP-Antwortstatuscode 200 gibt an, dass der Server betriebsbereit ist.

## Benutzer

Verwenden Sie die folgenden Endpunkte, um Benutzerinteraktionen zu verwalten.

### Themen

- [Anmeldung](#)
- [Überprüfen Sie das Benutzerkonto](#)
- [Melden Sie sich an](#)
- [Benutzerinformationen abrufen](#)
- [Auflisten aller Benutzer](#)
- [Delete user](#)
- [Abmelden](#)

### Anmeldung

- Unterstützte Methoden: POST
- Pfad: `/api/services/security/servicelogin/login`
- Argumente:
  - username
    - (erforderlich, Zeichenfolge)
  - password
    - (erforderlich, Zeichenfolge)
- Gibt den Benutzernamen und die Rollen des angemeldeten Benutzers zurück

### Beispielantwort

```
{"login":"some-user","roles":[{"id":0,"roleName":"ROLE_ADMIN"}]}
```

## Überprüfen Sie das Benutzerkonto

- Unterstützte Methoden: POST
- Pfad: `/api/services/security/servicelogin/hasAccount`
- Argumente: Keine
- Gibt Folgendes zurück: true, wenn der Benutzer bereits angemeldet ist

## Melden Sie sich an

- Unterstützte Methoden: POST
- Erfordert Authentifizierung und die Rolle `ROLE_ADMIN`.
- Pfad: `/api/services/security/servicelogin/recorduser`
- Argumente: Keine
- Gibt Folgendes zurück: true, wenn der Benutzer bereits angemeldet ist

## Benutzerinformationen abrufen

- Unterstützte Methoden: GET
- Pfad: `/api/services/security/servicelogin/userInfo`
- Argumente: Keine
- Gibt zurück: den Benutzernamen und die Rolle des aktuell verbundenen Benutzers

## Auflisten aller Benutzer

- Unterstützte Methoden: GET
- Erfordert Authentifizierung und die Rolle `ROLE_ADMIN`.
- Pfad: `/api/services/security/servicelogin/listusers`
- Argumente: Keine
- Gibt zurück: die Liste aller Benutzer

## Delete user

- Unterstützte Methoden: POST
- Pfad: `/api/services/security/servicelogin/deleteuser`



- Argumente:

login

(erforderlich, Zeichenfolge)

- Gibt Folgendes zurück: true, wenn der Benutzer erfolgreich entfernt wurde

#### Abmelden

- Unterstützte Methoden: POST
- Pfad: /api/services/security/servicelogout/logout
- Argumente: Keine
- Gibt Folgendes zurück: true, wenn der Benutzer erfolgreich abgemeldet wurde.

## JICS-Anwendungskonsole

Die JICS-Komponente ist die AWS Blu-Age-Unterstützung für die Modernisierung der älteren CICS-Ressourcen. Die Webanwendung JICS Application Console dient der Verwaltung von JICS-Ressourcen. Die folgenden Endpunkte ermöglichen die Ausführung der Verwaltungsaufgaben, ohne mit der JAC-Benutzeroberfläche interagieren zu müssen. Immer wenn ein Endpunkt eine Authentifizierung erfordert, muss die Anfrage Authentifizierungsdetails enthalten (in der Regel Benutzername/Passwort, wie es für die Standardauthentifizierung erforderlich ist). Endpunkte für die Webanwendung JICS Application Console verwenden den Stammpfad. /jac/

#### Themen

- [Verwaltung der JICS-Ressourcen](#)
- [Sonstige](#)
- [Endpunkte für die JAC-Benutzerverwaltung](#)

## Verwaltung der JICS-Ressourcen

Alle folgenden Endpunkte beziehen sich auf die JICS-Ressourcenverwaltung, sodass JICS-Administratoren täglich mit Ressourcen umgehen können.

#### Themen

- [JICS-LISTEN und -GRUPPEN auflisten](#)

- [Rufen Sie JICS-Ressourcen ab](#)
- [JICS-GRUPPEN auflisten](#)
- [Listet JICS-GRUPPEN für eine bestimmte LISTE auf](#)
- [LISTET JICS-Ressourcen für eine bestimmte GRUPPE AUF](#)
- [LISTET JICS-Ressourcen für eine bestimmte GRUPPE AUF \(Alternative mit einem Namen\)](#)
- [Bearbeitung der eigenen GRUPPEN mehrerer LISTEN](#)
- [Löscht eine LISTE](#)
- [Löscht eine GRUPPE](#)
- [Löschen Sie eine TRANSAKTION](#)
- [Löscht ein PROGRAMM](#)
- [Löscht eine DATEI](#)
- [Löscht eine TDQUEUE](#)
- [Löscht ein TSMODEL](#)
- [Elemente löschen](#)
- [Erstellen Sie eine LISTE](#)
- [Erstellen Sie eine GRUPPE](#)
- [Allgemeine Überlegungen zur Erstellung von RESSOURCEN](#)
- [Erstellen Sie eine TRANSAKTION](#)
- [Erstellen Sie ein PROGRAMM](#)
- [Erstellen Sie eine DATEI](#)
- [Erstellen Sie eine TDQUEUE](#)
- [Erstellen Sie ein TSMODEL](#)
- [Elemente erstellen](#)
- [Eine LISTE aktualisieren](#)
- [Aktualisieren Sie eine GRUPPE](#)
- [Allgemeine Überlegungen zur Aktualisierung von RESSOURCEN](#)
- [Eine TRANSAKTION aktualisieren](#)
- [Aktualisieren Sie ein PROGRAMM](#)
- [Eine DATEI aktualisieren](#)

- [Aktualisieren Sie eine TDQUEUE](#)
- [Aktualisieren Sie ein TSMODEL](#)
- [Elemente aktualisieren](#)
- [Elemente vergrößern](#)
- [Ruft Elemente ab](#)
- [JICS CRUD-Vorgang](#)

## JICS-LISTEN und -GRUPPEN auflisten

LIST und GROUPS sind die Haupteigentümer der Container-Ressourcen innerhalb der JICS-Komponente. Alle JICS-Ressourcen müssen zu einer GRUPPE gehören. Gruppen können zu LISTEN gehören, dies ist jedoch nicht zwingend erforderlich. LISTEN existieren möglicherweise in einer bestimmten JICS-Umgebung nicht einmal, aber in den meisten Fällen sind LISTEN dazu da, eine zusätzliche Organisationsebene für Ressourcen zu schaffen. Weitere Informationen zur Organisation der CICS-Ressourcen finden Sie unter [CICS-Ressourcen](#).

- Unterstützte Methode: GET
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/listJicsListsAndGroups
- Argumente: Keine
- Gibt Folgendes zurück: eine Liste serialisierter JicsContainer Objekte, sowohl LISTS als auch GROUPS, als JSON.

### Beispielantwort:

```
[
  {
    "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
}
]
```

Rufen Sie JICS-Ressourcen ab

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/retrieveJicsResources`
- Argumente: Eine JSON-Nutzlast, die die JICS-Ressourcen darstellt, die Sie abrufen möchten. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfective.bluage.jac.entities.request.RetrieveOperationRequest`
- Rückgabe: Eine Liste serialisierter Objekte. JicsResource Die Objekte werden in keiner bestimmten Reihenfolge zurückgegeben und sind unterschiedlichen Typs, wie PROGRAM, TRANSACTION, FILE usw.

JICS-GRUPPEN auflisten

- Unterstützte Methode: GET
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/listJicsGroups`
- Gibt zurück: eine Liste serialisierter JicsContainer Objekte (GROUPS) als JSON. Die GROUPS werden ohne ihre eigenen LIST-Informationen zurückgegeben.

**Beispielantwort:**

```
[
  {
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  },
  {
    "jacType": "JACGroup",
    "name": "TEST",
    "isActive": true,
    "children": []
  }
]
```

Listet JICS-GRUPPEN für eine bestimmte LISTE auf

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/listGroupsForList
- Argumente: eine JSON-Payload, die die JICS-LISTE darstellt, nach deren GRUPPEN wir suchen. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfective.bluage.jac.entities.JACList`

**Beispielanfrage:**

```
{
  "jacType": "JACList",
  "name": "MURACHS",
  "isActive": true
}
```

- Gibt zurück: eine Liste von serialisierten JicsContainer Objekten (GROUPS) als JSON, die an die angegebene LISTE angehängt sind. Die GROUPS werden ohne ihre eigenen LIST-Informationen zurückgegeben.

**Beispielantwort:**

```
[
  {
    "jacType": "JACGroup",
    "name": "MURACHS",
    "isActive": true,
    "children": []
  }
]
```

## LISTET JICS-Ressourcen für eine bestimmte GRUPPE AUF

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/listResourcesForGroup
- Argumente: eine JSON-Nutzlast, die die JICS-GRUPPE darstellt, nach deren Ressourcen wir suchen. Dies ist die JSON-Serialisierung eines Objekts.  
com.netfactive.bluage.jac.entities.JACGroup Sie müssen nicht alle Felder für die GRUPPE angeben, aber der Name ist obligatorisch.

Musteranfrage:

```
{
  "jacType":"JACGroup",
  "name":"MURACHS",
  "isActive":true
}
```

- Gibt zurück: eine Liste serialisierter JicsResource Objekte, die der angegebenen GRUPPE gehören. Die Objekte werden in keiner bestimmten Reihenfolge zurückgegeben und sind unterschiedlichen Typs (PROGRAM, TRANSACTION, FILE, etc...).

## LISTET JICS-Ressourcen für eine bestimmte GRUPPE AUF (Alternative mit einem Namen)

- Unterstützte Methode: POST
- Erfordert Authentifizierung
- Pfad: /api/services/rest/jicsservice/listResourcesForGroupName

- Argumente: Der Name der GRUPPE, der die Ressourcen gehören, nach denen wir suchen.
- Gibt zurück: eine Liste serialisierter JicsResource Objekte, die der angegebenen GRUPPE gehören. Die Objekte werden in keiner bestimmten Reihenfolge zurückgegeben und sind unterschiedlichen Typs (PROGRAM, TRANSACTION, FILE, etc...)

### Bearbeitung der eigenen GRUPPEN mehrerer LISTEN

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/editGroupsList
- Argumente: eine JSON-Repräsentation einer Sammlung von LISTEN mit untergeordneten GRUPPEN;

#### Musteranfrage:

```
[
  {
    "jacType": "JACList",
    "name": "MURACHS",
    "isActive": true,
    "children": [
      {
        "jacType": "JACGroup",
        "name": "MURACHS",
        "isActive": true,
        "children": []
      },
      {
        "jacType": "JACGroup",
        "name": "TEST",
        "isActive": true,
        "children": []
      }
    ]
  }
]
```

Vor dieser Bearbeitung gehörte nur die Gruppe mit dem Namen „MURACHS“ zur LISTE mit dem Namen „MURACHS“. Mit dieser Bearbeitung „fügen“ wir die Gruppe mit dem Namen „TEST“ zur LISTE mit dem Namen „MURACHS“ hinzu.

- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurden die LIST-Änderungen ordnungsgemäß im zugrunde liegenden JICS-Speicher gespeichert.

#### Löscht eine LISTE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteList`
- Argumente: eine JSON-Payload, die die zu löschende JICS-LISTE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACList`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurden die LIST-Löschungen auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

#### Löscht eine GRUPPE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteGroup`
- Argumente: eine JSON-Nutzlast, die die zu löschende JICS-GRUPPE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACGroup`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurden die GROUP-Löschvorgänge auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

#### Löschen Sie eine TRANSAKTION

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteTransaction`



- Argumente: eine JSON-Nutzlast, die die zu löschende JICS-Transaktion darstellt. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfactive.bluage.jac.entities.JACTransaction`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurden die TRANSACTION-Löschvorgänge auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

### Löscht ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteProgram`
- Argumente: eine JSON-Payload, die das zu löschende JICS-Programm darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfactive.bluage.jac.entities.JACProgram`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das Löschen von PROGRAMM auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

### Löscht eine DATEI

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteFile`
- Argumente: eine JSON-Nutzlast, die die zu löschende JIC-Datei darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfactive.bluage.jac.entities.JACFile`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das Löschen von DATEIEN auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

### Löscht eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteTDQueue`

- Argumente: eine JSON-Payload, die die zu löschende JICS-TDQUEUE darstellt. Dies ist die JSON-Serialisierung eines ``com.neffective.bluage.jac.entities.JACTDQueue``-Objekts.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurden die TDQUEUE-Löschvorgänge auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

#### Löscht ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteTSMODEL`
- Argumente: eine JSON-Payload, die das zu löschende JICS TSMODEL darstellt. Dies ist die JSON-Serialisierung eines ``com.neffective.bluage.jac.entities.JactsModel``-Objekts.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TSMODEL-Löschung auf dem zugrunde liegenden JICS-Speicher ordnungsgemäß ausgeführt.

#### Elemente löschen

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/deleteElements`
- Argumente: Eine JSON-Nutzlast, die die zu löschenden JICS-Elemente darstellt.
- Gibt einen booleschen Wert zurück, der true angibt, dass der Löschvorgang im zugrunde liegenden JICS-Speicher erfolgreich durchgeführt wurde.

#### Erstellen Sie eine LISTE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/createList`
- Argumente: eine JSON-Payload, die die zu erstellende JICS-LISTE darstellt. Dies ist die JSON-Serialisierung eines ``com.neffective.bluage.jac.Entities.JacList``-Objekts.

- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die LISTE ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

#### Note

Die LISTE wird immer leer erstellt. Das Anhängen von GRUPPEN an die LISTE erfordert einen weiteren Vorgang.

## Erstellen Sie eine GRUPPE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/createGroup`
- Argumente: eine JSON-Nutzlast, die die zu erstellende JICS-GRUPPE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfactive.bluage.jac.entities.JACGroup`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die GRUPPE ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

#### Note

Die GRUPPE wird immer leer erstellt. Das Anhängen von RESSOURCEN an die GRUPPE erfordert zusätzliche Operationen (beim Erstellen von Ressourcen werden sie automatisch an eine bestimmte GRUPPE angehängt).

## Allgemeine Überlegungen zur Erstellung von RESSOURCEN

Alle folgenden Endpunkte beziehen sich auf die Erstellung von JICS-RESSOURCEN und weisen einige gemeinsame Einschränkungen auf: In der Payload der Anforderung, die an den Endpunkt gesendet werden soll, muss das `groupName` Feld einen Wert haben.

### Eigentumsbeschränkung für GRUPPEN:

Keine Ressource kann erstellt werden, ohne an eine bestehende Gruppe angehängt zu sein, und der Endpunkt verwendet den `groupName`, um die Gruppe abzurufen, an die diese Ressource

angehängt wird. Der `groupName` muss auf den Namen einer vorhandenen GRUPPE verweisen. Eine Fehlermeldung mit dem HTTP-STATUS 400 wird gesendet, wenn sie nicht auf eine bestehende Gruppe im zugrunde liegenden JICS-Speicher verweist. `groupName`

Unicity-Einschränkung innerhalb einer GRUPPE:

Eine bestimmte Ressource mit einem bestimmten Namen muss innerhalb einer bestimmten Gruppe eindeutig sein. Die Prüfung auf Eindeutigkeit wird von jedem Endpunkt der Ressourcenerstellung durchgeführt. Wenn die angegebene Nutzlast die Unicity-Beschränkung nicht einhält, sendet der Endpunkt eine Antwort mit dem HTTP-STATUS 400 (BAD REQUEST) — siehe die Beispielantwort unten.

Beispielnutzlast: Wir versuchen, die Transaktion 'ARIT' in der Gruppe 'TEST' zu erstellen, aber eine Transaktion mit diesem Namen existiert bereits in dieser GROUP.

```
{
  "jacType":"JACTransaction",
  "name":"ARIT",
  "groupName":"TEST",
  "isActive":true
}
```

Wir haben die folgende Fehlerantwort erhalten:

```
{
  "timestamp": 1686759054510,
  "status": 400,
  "error": "Bad Request",
  "path": "/jac/api/services/rest/jicsservice/createTransaction"
}
```

Durch die Überprüfung der Serverprotokolle wird der Ursprung des Problems bestätigt:

```
2023-06-14 18:10:54 default          TRACE - o.s.w.m.HandlerMethod
      - Arguments: [java.lang.IllegalArgumentException: Transaction already
present in the group, org.springframework.security.web.header.HeaderWriterFilter
$HeaderWriterResponse@e34f6b8]
2023-06-14 18:10:54 default          ERROR - c.n.b.j.a.WebConfig          -
400
```

```
java.lang.IllegalArgumentException: Transaction already present in the group
at
com.netfactive.bluage.jac.server.services.rest.impl.JicsServiceImpl.createElement(JicsServiceI
```

## Erstellen Sie eine TRANSAKTION

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/createTransaction
- Argumente: eine JSON-Nutzlast, die die zu erstellende JICS-TRANSAKTION darstellt. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfactive.bluage.jac.entities.JACTransaction`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TRANSACTION ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

## Erstellen Sie ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/createProgram
- Argumente: eine JSON-Payload, die das zu erstellende JICS-PROGRAMM darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfactive.bluage.jac.entities.JACProgram`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das PROGRAMM ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

## Erstellen Sie eine DATEI

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/createFile

- Argumente: eine JSON-Nutzlast, die die zu erstellende JICS-DATEI darstellt. Dies ist die JSON-Serialisierung eines ``com.netfective.bluage.jac.Entities.JacFile``-Objekts.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die DATEI ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

#### Erstellen Sie eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Pfad: `/api/services/rest/jicsservice/createTDQueue`
- Argumente: eine JSON-Payload, die die zu erstellende JICS-TDQUEUE darstellt. Dies ist die JSON-Serialisierung eines ``com.netfective.bluage.jac.entities.JACTDQueue``-Objekts.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TDQUEUE ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

#### Erstellen Sie ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Pfad: `/api/services/rest/jicsservice/createTSMODEL`
- Argumente: eine JSON-Nutzlast, die das zu erstellende JICS TSMODEL darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACTSMODEL`
- Gibt einen booleschen Wert zurück, der `true` angibt, dass die Erstellung von Elementen im zugrunde liegenden JICS-Speicher erfolgreich durchgeführt wurde.

#### Elemente erstellen

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Pfad: `/api/services/rest/jicsservice/createElements`
- Argumente: eine JSON-Nutzlast, die die zu erstellenden JICS-Elemente darstellt.

- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das TSMODEL ordnungsgemäß im zugrunde liegenden JICS-Speicher erstellt.

### Eine LISTE aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateList`
- Argumente: eine JSON-Payload, die die zu aktualisierende JICS-LISTE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACList`  
Es ist nicht erforderlich, die untergeordneten Elemente der LIST anzugeben, der LIST-Aktualisierungsmechanismus berücksichtigt dies nicht.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die LISTE im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

Wenn das LIST-Flag 'isActive' aktualisiert wird, wird dies auf alle eigenen Elemente der LISTE übertragen, d. h. auf alle GRUPPEN, die der LISTE gehören, und auf alle RESSOURCEN, die diesen GRUPPEN gehören. Dies ist eine bequeme Methode, um viele Ressourcen mit einem einzigen Vorgang über mehrere GRUPPEN hinweg zu deaktivieren.

### Aktualisieren Sie eine GRUPPE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateGroup`
- Argumente: eine JSON-Payload, die die zu aktualisierende JICS-GRUPPE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACGroup`  
Es ist nicht erforderlich, die untergeordneten Elemente der GRUPPE anzugeben, der GROUP-Aktualisierungsmechanismus berücksichtigt dies nicht.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die GRUPPE im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

**Note**

Durch die Aktualisierung des GROUP-Flags „isActive“ wird dies auf alle eigenen Elemente der GRUPPE übertragen, d. h. auf alle RESSOURCEN, die der GRUPPE gehören. Dies ist eine bequeme Methode, um viele Ressourcen mit einem einzigen Vorgang innerhalb einer bestimmten GRUPPE zu deaktivieren.

## Allgemeine Überlegungen zur Aktualisierung von RESSOURCEN

Bei allen folgenden Endpunkten geht es um die Aktualisierung von JICS-RESSOURCEN. Mithilfe des `groupName` Felds können Sie die Eigentümergruppe einer beliebigen JICS-RESSOURCE ändern, sofern der Feldwert auf eine bestehende GRUPPE im zugrunde liegenden JICS-Speicher verweist (andernfalls erhalten Sie vom Endpunkt eine BAD REQUEST-Antwort (HTTP STATUS 400)).

### Eine TRANSAKTION aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Pfad: `/api/services/rest/jicsservice/updateTransaction`
- Argumente: eine JSON-Nutzlast, die die zu aktualisierende JICS-TRANSAKTION darstellt. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfective.bluage.jac.entities.JACTransaction`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TRANSACTION im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

### Aktualisieren Sie ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: `ROLE_ADMIN`, `ROLE_SUPER_ADMIN`, `ROLE_USER`
- Pfad: `/api/services/rest/jicsservice/updateProgram`
- Argumente: eine JSON-Payload, die das zu aktualisierende JICS-PROGRAMM darstellt. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfective.bluage.jac.entities.JACProgram`



- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das PROGRAMM im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

#### Eine DATEI aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateFile`
- Argumente: eine JSON-Nutzlast, die die zu aktualisierende JICS-DATEI darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACFile`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die DATEI im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

#### Aktualisieren Sie eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateTDQueue`
- Argumente: eine JSON-Payload, die die zu aktualisierende JICS-TDQUEUE darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACTDQueue`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TDQueue im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

#### Aktualisieren Sie ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateTSModel`
- Argumente: eine JSON-Nutzlast, die das zu aktualisierende JICS TSMODEL darstellt. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACTSModel`

- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das TSMODEL im zugrunde liegenden JICS-Speicher ordnungsgemäß aktualisiert.

#### Elemente aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/updateElements`
- Argumente: Eine JSON-Nutzlast, die die zu aktualisierenden Elemente darstellt.
- Gibt einen booleschen Wert zurück, der `true` angibt, dass die Aktualisierung der Elemente im zugrunde liegenden JICS-Speicher erfolgreich durchgeführt wurde.

#### Elemente vergrößern

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/upsertElements`
- Argumente: Eine JSON-Nutzlast, die die Elemente darstellt, die aktualisiert werden sollen.
- Gibt einen booleschen Wert zurück, der `true` angibt, dass das Upsert der Elemente erfolgreich im zugrunde liegenden JICS-Speicher ausgeführt wurde.

#### Ruft Elemente ab

- Unterstützte Methode: GET
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: `/api/services/rest/jicsservice/retrieveElements`
- Argumente: Keine
- Gibt eine Liste aller serialisierten JICS-Ressourcen zurück.

## JICS CRUD-Vorgang

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die folgenden Rollen: ROLE\_ADMIN, ROLE\_SUPER\_ADMIN, ROLE\_USER
- Pfad: /api/services/rest/jicsservice/jicsCrudOperation
- Argumente: eine JSON-Nutzlast, die die JICS-Ressourcen darstellt, nach denen wir suchen. Dies ist die JSON-Serialisierung eines Objekts.  
`com.netfactive.bluage.jac.entities.request.JicsCrudOperationRequest`
- Gibt eine JSON-Nutzlast zurück, die die Antwort darstellt. Dies ist die JSON-Serialisierung eines `com.netfactive.bluage.jac.entities.request.JicsCrudOperationResponse` Objekts.

## Sonstige

### Themen

- [Integritätsstatus des JICS-Servers](#)

### Integritätsstatus des JICS-Servers

- Unterstützte Methode: GET
- Pfad: /api/services/rest/jicsserver/serverIsUp
- Argumente: Keine
- Gibt zurück: Keine. Eine HTTP STATUS 200-Antwort zeigt an, dass der Server betriebsbereit ist.

## Endpunkte für die JAC-Benutzerverwaltung

Verwenden Sie die folgenden Endpunkte, um Benutzerinteraktionen zu verwalten.

### Themen

- [Einen Benutzer protokollieren](#)
- [Es wird getestet, ob mindestens ein Benutzer im System existiert](#)
- [Einen neuen Benutzer aufzeichnen](#)
- [Informationen zum Benutzer](#)

- [Benutzer auflisten](#)
- [Löschen eines Benutzers](#)
- [Melden Sie den aktuellen Benutzer ab](#)

### Einen Benutzer protokollieren

- Unterstützte Methode: POST
- Pfad: `/api/services/security/servicelogin/login`
- Argumente: Keine
- Gibt die JSON-Serialisierung eines `com.netfactive.bluage.jac.entities.SignOn` Objekts zurück, die den Benutzer darstellt, dessen Anmeldeinformationen in der aktuellen Anfrage angegeben wurden. Das Passwort ist in der Ansicht des zurückgegebenen Objekts verborgen. Die Rollen, die dem Benutzer zugewiesen wurden, werden aufgelistet.

### Beispielantwort:

```
{
  "login": "some-admin",
  "password": null,
  "roles": [
    {
      "id": 0,
      "roleName": "ROLE_ADMIN"
    }
  ]
}
```

Es wird getestet, ob mindestens ein Benutzer im System existiert

- Unterstützte Methode: GET
- Pfad: `/api/services/security/servicelogin/hasAccount`
- Argumente: Keine
- Gibt den booleschen Wert zurück, `true` wenn mindestens ein anderer Benutzer als der Standard-Super-Admin-Benutzer erstellt wurde. Gibt andernfalls zurück `false`.

## Einen neuen Benutzer aufzeichnen

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die Rolle ROLE\_ADMIN.
- Pfad: `/api/services/security/servicelogin/recorduser`
- Argumente: Die JSON-Serialisierung eines `com.netfective.bluage.jac.entities.SignOn` Objekts, das den Benutzer darstellt, der dem Speicher hinzugefügt werden soll. Die Rollen für den Benutzer sollten definiert werden, andernfalls kann der Benutzer die JAC-Einrichtung und die Endpunkte möglicherweise nicht verwenden.
- Gibt den booleschen Wert zurück, `true` wenn der Benutzer erfolgreich erstellt wurde. Gibt andernfalls zurück `false`.

### Musteranfrage:

```
{
  "login": "simpleuser",
  "password": "simplepassword",
  "roles": [
    {
      "id": 2,
      "roleName": "ROLE_USER"
    }
  ]
}
```

Bei der Aufzeichnung eines neuen Benutzers können nur die folgenden Rollen verwendet werden:

- ROLE\_ADMIN: kann JICS-Ressourcen und -Benutzer verwalten.
- ROLE\_USER: kann JICS-Ressourcen verwalten, aber keine Benutzer.

### Informationen zum Benutzer

- Unterstützte Methode: GET
- Erfordert Authentifizierung und ADMIN-Rechte
- Pfad: `/api/services/security/servicelogin/userInfo`
- Gibt den Benutzernamen und die Rollen des aktuell verbundenen Benutzers zurück.

## Benutzer auflisten

- Unterstützte Methode: GET
- Erfordert Authentifizierung und die Rolle `ROLE_ADMIN`.
- Pfad: `/api/services/security/servicelogin/listusers`
- Argumente: Keine
- Gibt eine Liste von `zurückcom.netfective.bluage.jac.entities.SignOn`, serialisiert als JSON.

## Löschen eines Benutzers

- Unterstützte Methode: POST
- Pfad: `/api/services/security/servicelogin/deleteuser`
- Argumente: Die JSON-Serialisierung eines `com.netfective.bluage.jac.entities.SignOn` Objekts, das den Benutzer darstellt, der aus dem Speicher entfernt werden soll.
- Gibt den booleschen Wert zurück, `true` wenn der Benutzer erfolgreich entfernt wurde.

### Important

Diese Aktion kann nicht rückgängig gemacht werden. Der gelöschte Benutzer kann sich nicht wieder mit der JAC-Anwendung verbinden.

## Melden Sie den aktuellen Benutzer ab

- Unterstützte Methode: GET
- Pfad: `/api/services/security/servicelogout/logout`
- Argumente: Keine
- Gibt die JSON-Nachricht zurück, `{"success":true}` wenn der aktuelle Benutzer erfolgreich abgemeldet wurde. Die zugehörige HTTP-Sitzung wird ungültig gemacht.

## Datenstrukturen

In diesem Abschnitt werden Einzelheiten zu den verschiedenen Datenstrukturen beschrieben.

## Themen

- [Nachrichtenstruktur mit Details zur Jobausführung](#)
- [Struktur der Ergebnisse des Transaktionsstarts](#)
- [Aufbau des Transaktionsstarts, Aufzeichnung der Ergebnisse](#)
- [Möglicher Status eines Jobs in einer Warteschlange](#)
- [Job abschicken und Job-Eingabe planen](#)
- [Liste der Antworten auf geplante Jobs](#)
- [Liste der Antworten auf wiederkehrende Jobs](#)

## Nachrichtenstruktur mit Details zur Jobausführung

Die Details zur Auftragsausführung enthalten die folgenden Felder:

### ScriptID

die Kennung des aufgerufenen Skripts.

### Anrufer

IP-Adresse des Anrufers.

### Bezeichner

eindeutige Kennung für die Auftragsausführung.

### startTime

Datum und Uhrzeit des Beginns der Auftragsausführung.

### endTime

Datum und Uhrzeit, an dem die Auftragsausführung beendet wurde.

### Status

ein Status für die Auftragsausführung. Ein möglicher Wert unter:

- DONE: Die Ausführung des Jobs wurde normal beendet.
- TRIGGERED: Die Auftragsausführung wurde ausgelöst, aber noch nicht gestartet.
- RUNNING: Die Jobausführung läuft.

- KILLED: Die Auftragsausführung wurde beendet.
- FAILED: Die Auftragsausführung ist fehlgeschlagen.

## Ausführungsergebnis

eine Meldung, die das Ergebnis der Jobausführung zusammenfasst. Diese Nachricht kann entweder eine einfache Nachricht sein, falls die Jobausführung noch nicht abgeschlossen ist, oder eine JSON-Struktur mit den folgenden Feldern:

- ExitCode: numerischer Exit-Code; negative Werte deuten auf Fehlersituationen hin.
- Programm: das neueste Programm, das durch den Job gestartet wurde.
- Status: ein möglicher Wert unter:
  - Error: wenn ExitCode = -1; dies entspricht einem (technischen) Fehler, der während der Jobausführung auftritt.
  - Failed: wenn exitcode = -2; Dies entspricht einem Fehler, der während der Ausführung eines Serviceprogramms auftritt (wie eine ABEND-Situation).
  - Succeeded: wenn ExitCode  $\geq$  0;
- stepName: Name des letzten Schritts, der im Job ausgeführt wurde.

## Ausführungsmodus

entweder SYNCHRON oder ASYNCHRON, je nachdem, wie der Job gestartet wurde.

## Beispielausgabe:

```
{
  "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"
}
```

## Struktur der Ergebnisse des Transaktionsstarts

Die Struktur kann die folgenden Felder enthalten:



## Ergebnis

eine Zeichenfolge, die das Ergebnis der Transaktionsausführung darstellt. Die möglichen Werte sind:

- **Success**: Die Ausführung der Transaktion wurde ordnungsgemäß abgeschlossen.
- **Failure**: Die Transaktionsausführung konnte nicht ordnungsgemäß beendet werden, es sind einige Probleme aufgetreten.

## Komma

eine Zeichenfolge, die den COMMAREA-Endwert als Byte64-codiertes Bytearray darstellt. Könnte eine leere Zeichenfolge sein.

## ContainerRecord

(optional) eine Zeichenfolge, die den Datensatzinhalt des CONTAINERS als Byte64-codiertes Bytearray darstellt.

## Beschreibung des Servers

Kann Informationen über den Server enthalten, der die Anfrage bearbeitet hat (zu Debugging-Zwecken). Könnte eine leere Zeichenfolge sein.

## Ein Biegecode

(optional) Wenn das Programm, auf das von der gestarteten Transaktion verwiesen wurde, eine Änderung vorgenommen hat, wird der Wert des Abendcodes als Zeichenfolge in diesem Feld zurückgegeben.

## Beispielantworten:

### Herzlichen Glückwunsch

```
{
  "outcome": "Success",
  "commarea": "",
  "serverDescription": ""
}
```

## Fehler

```
{
  "outcome": "Failure",
```

```
"commarea": "",  
"serverDescription": "",  
"abendCode": "AEIA"  
}
```

## Aufbau des Transaktionsstarts, Aufzeichnung der Ergebnisse

Die Struktur kann die folgenden Felder enthalten:

### Inhalt aufzeichnen

eine Zeichenfolge, die den Inhalt des COMMAREA-Datensatzes als Byte64-codiertes Bytearray darstellt.

### Container-Datensatz

eine Zeichenfolge, die den Datensatzinhalt des CONTAINERS als Byte64-codiertes Bytearray darstellt.

### Beschreibung des Servers

Kann Informationen über den Server enthalten, der die Anfrage bearbeitet hat (zu Debugging-Zwecken). Könnte eine leere Zeichenfolge sein.

### Beispielantworten:

#### Herzlichen Glückwunsch

```
{  
  "recordContent": "",  
  "serverDescription": ""  
}
```

## Möglicher Status eines Jobs in einer Warteschlange

In einer Warteschlange können Jobs den folgenden Status haben:

### ACTIVE

Der Job wird derzeit in der Warteschlange ausgeführt.

### EXECUTION\_WAIT

Der Job wartet darauf, dass ein Thread verfügbar ist.

## GEPLANT

Jobs sind für die Ausführung an einem bestimmten Datum und zu einer bestimmten Uhrzeit geplant.

## HOLD

Der Job wartet darauf, veröffentlicht zu werden, bevor er ausgeführt wird.

## COMPLETED

Der Job wurde erfolgreich ausgeführt.

## FEHLGESCHLAGEN

Die Ausführung des Jobs ist fehlgeschlagen.

## UNKNOWN

Der Status ist unbekannt.

## Job abschicken und Job-Eingabe planen

Die Eingabe „Job senden“ und „Job planen“ ist die JSON-Serialisierung eines `com.netfactive.bluage.gapwalk.rt.jobqueue.SubmitJobMessage` Objekts. Die folgende Beispieleingabe zeigt alle Felder für eine solche Bohne.

Beispieleingabe für den Job „Auftrag einreichen“:

```
{
  "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
}
```

## Beispieleingabe für einen Job planen:

```
{
  "scheduleCron": "*/2 * * * * ?",
  "programName": "LOGPGM",
  "programParams": {
    "cl_sbmjob_param_json": "[\"./output/schedule-job-log.txt\", \"Every 2
seconds!\"]"
  },
  "localDataAreaValue": "",
  "userName": "PV0",
  "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"
}
```

### Nummer des Auftrags

Wenn die Auftragsnummer 0 ist, wird die Auftragsnummer automatisch anhand der nächsten Nummer in der Auftragsnummernfolge generiert. Dieser Wert sollte auf 0 gesetzt werden (außer zu Testzwecken).

### Priorität des Auftrags

Die Standard-Auftragspriorität in AS400 ist 5. Der gültige Bereich liegt zwischen 0 und 9, wobei 0 die höchste Priorität darstellt.

### Beruf OnHold

Wenn ein Job in der Warteschleife eingereicht wird, wird er nicht sofort ausgeführt, sondern erst, wenn ihn jemand „freigibt“. Ein Job kann mit der REST-API (/release oder /release-all) veröffentlicht werden.

### ScheduleDate und ScheduleTime

Wenn diese Werte nicht Null sind, wird der Job am angegebenen Datum und zur angegebenen Uhrzeit ausgeführt.

### Datum

Kann mit dem Format mmDyy oder ddMMyyyy angegeben werden (die Größe der Eingabe bestimmt, welches Format verwendet wird)

## Zeit

Kann mit dem Format HHmm oder HHmmss geliefert werden (die Größe der Eingabe bestimmt, welches Format verwendet wird)

## ProgramParams

Wird als Map an das Programm übergeben.

## Zeitplan MisfirePolicy

Definiert die Strategie, die verwendet wird, wenn ein Trigger fehlausgelöst wird. Die folgenden Werte sind möglich:

1. Löst die erste Fehlzündung aus und verwirft die anderen Fehlzündungen.
2. Reichen Sie einen Job ein, der wegen der ersten Fehlzündung zurückgestellt wurde, und verwerfen Sie die anderen Fehlzündungen.
3. Verwerfen Sie die Fehlzündung.
4. Löse alle Fehlzündungen. In der Job-Warteschlange werden alle Jobs ausgeführt.

## Liste der Antworten auf geplante Jobs

Dies ist die Struktur des Endpunkts der Jobwarteschlange mit der Liste der Jobs. Die Nachricht zum Absenden des Jobs, die zum Senden dieses Jobs verwendet wurde, ist Teil der Antwort. Dies kann zur Nachverfolgung oder zum Testen und erneuten Senden verwendet werden. Wenn ein Job abgeschlossen ist, werden auch das Start- und Enddatum eingetragen.

```
[
  {
    "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
}
]

```

## Liste der Antworten auf wiederkehrende Jobs

Dies ist die Struktur des Auftragswarteschlangenendpunkts `/schedule/list`.

```

[
  {
    "id": 1,
    "status": "ACTIVE",
    "jobNumber": 1,
    "userName": "PV0",
    "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": "PV0",
      "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 Einrichtung von Blu Age Runtime (nicht verwaltet)

In diesem Abschnitt werden die Schritte zur Einrichtung von AWS Blu Age Runtime (nicht verwaltet) auf Ihrer AWS Infrastruktur erläutert.

### Themen

- [AWS Voraussetzungen für die Laufzeit von Blu Age](#)
- [AWS Einführung in Blu Age Runtime](#)
- [Anforderungen an die Einrichtung der Infrastruktur für AWS Blu Age Runtime \(nicht verwaltet\)](#)
- [AWS Bereitstellung von Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)
- [AWS Bereitstellung von Blu Age Runtime auf Amazon EC2](#)
- [Testen Sie die Anwendung PlanetsDemo](#)

## AWS Voraussetzungen für die Laufzeit von Blu Age

AWS Blu Age Runtime (nicht verwaltet) ist in verschiedenen [Release-Versionen](#) erhältlich. Wenn Sie laufende Modernisierungsprojekte haben, benötigen Sie möglicherweise inkrementelle Versionen der Runtime für Implementierungs- und Testzwecke. Um Ihre Bedürfnisse zu definieren, wenden Sie sich an Ihren AWS Blu Age Delivery Manager.

Bevor Sie mit dem Onboarding-Prozess für AWS Blu Age Runtime (nicht verwaltet) beginnen, gehen Sie wie folgt vor:

- Vergewissern Sie sich, dass Sie ein AWS Konto haben.
- Stellen Sie sicher, dass Sie über eine modernisierte Anwendung verfügen, die mit Blu Age überarbeitet wurde. AWS
- Wählen Sie eine AWS Region und eine der Rechenoptionen, die für AWS Blu Age Runtime (nicht verwaltet) unterstützt werden.
- Wählen Sie die AWS Blu Age Runtime-Version aus, die Sie verwenden möchten.



- Überprüfen [the section called “Anforderungen an die Einrichtung der Infrastruktur”](#) und validieren Sie die zusätzlichen Komponenten, die für die Ausführung der AWS Blu Age Runtime (nicht verwaltet) erforderlich sind.

#### Note

[Wenn Sie die Funktionen von AWS Blu Age Runtime \(nicht verwaltet\) testen möchten, können Sie die Demo-Anwendung verwenden Planets Demo, die Sie von PlanetsDemo - v1.zip herunterladen können.](#)

## AWS Einführung in Blu Age Runtime

Erstellen Sie zunächst einen AWS Support Fall, um das Onboarding für den Zugriff auf AWS Blu Age Runtime zu beantragen. Geben Sie in Ihrer Anfrage Ihre AWS-Konto ID, die AWS Region, die Sie verwenden möchten, sowie eine Rechenoption und eine Runtime-Version an. Wenn Sie sich nicht sicher sind, welche Version Sie benötigen, wenden Sie sich an Ihren AWS Blu Age-Liefermanager.

#### Note

Die AWS Blu Age Runtime ist in zwei Hauptvarianten erhältlich: Alpha-Vorabversionen und offizielle Versionen. Informationen darüber, welche Version Sie verwenden sollten, finden Sie unter [Erste Schritte](#) auf der Blu Insights-Website oder wenden Sie sich an Ihren AWS Blu Age-Liefermanager.

## AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2

Wir speichern die (nicht verwalteten) AWS Blu Age Runtime-Artefakte in verschiedenen Amazon S3 S3-Buckets, je nach Region und Rechenauswahl. Verwenden Sie den in der folgenden Tabelle aufgeführten Namen, um auf den Bucket AWS-Region für Ihre für AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2 zuzugreifen.

#### Note

Diese Tabelle gilt sowohl für Amazon EC2- als auch für Amazon EC2 EC2-Instances, die in Amazon ECS und Amazon EKS verwendet werden.

AWS-Region	Bucket veröffentlichen	Bucket vor der Veröffentlichung
USA Ost (Ohio)	aws-bluage-runtime-artefact s-055777665268-us-east-2	aws-bluage-runtime-artifacts- dev-055777665268-us-east-2
USA Ost (Nord-Virginia)	aws-bluage-runtime-artefact s-139023371234-us-east-1	aws-bluage-runtime-artifacts- dev-139023371234-us-east-1
USA West (Nordkalifornien)	aws-bluage-runtime-artefact s-788454048782-us-west-1	aws-bluage-runtime-artifacts- dev-788454048782-us-west-1
USA West (Oregon)	aws-bluage-runtime-artefact s-836771190483-us-west-2	aws-bluage-runtime-artifacts- dev-836771190483-us-west-2
Europa (Irland)	aws-bluage-runtime-artefact s-925278190477-eu-west-1	aws-bluage-runtime-artefacts- dev-925278190477-eu-west-1
Europa (Paris)	aws-bluage-runtime-artefact s-673009995881-eu-west-3	aws-bluage-runtime-artefacts- dev-673009995881-eu-west-3
Europa (Frankfurt)	aws-bluage-runtime-artefact s-485196800481-eu-central-1	aws-bluage-runtime-artefacts- dev-485196800481-eu-centr al-1
Südamerika (São Paulo)	aws-bluage-runtime-artefact s-737536804457-sa-east-1	aws-bluage-runtime-artifacts- dev-737536804457-sa-east-1
Asien-Pazifik (Tokio)	aws-bluage-runtime-artefact s-445578176276-ap-northeast -1	aws-bluage-runtime-artifacts- dev-445578176276-ap-north east-1
Asien-Pazifik (Sydney)	aws-bluage-runtime-artefact s-726160321909-ap-southeast -2	aws-bluage-runtime-artifacts- dev-726160321909-ap-south east-2

## AWS Blu Age Runtime (nicht verwaltet) auf Amazon ECS, verwaltet von Fargate

Wir speichern die (nicht verwalteten) AWS Blu Age Runtime-Artefakte in verschiedenen Amazon S3 S3-Buckets, je nach Region und Rechenauswahl. Verwenden Sie den in der folgenden Tabelle aufgeführten Namen, um auf den Bucket AWS-Region für Ihre von Fargate verwaltete AWS Blu Age Runtime (nicht verwaltet) auf Amazon ECS zuzugreifen.

AWS-Region	Bucket veröffentlichen	Bucket vor der Veröffentlichung
USA Ost (Ohio)	aws-bluage-runtime-fargate-rel-483416914331-us-east-2	aws-bluage-runtime-fargate-dev-483416914331-us-east-2
USA Ost (Nord-Virginia)	aws-bluage-runtime-fargate-rel-308472162679-us-east-1	aws-bluage-runtime-fargate-dev-308472162679-us-east-1
USA West (Nordkalifornien)	aws-bluage-runtime-fargate-rel-343763094578-us-west-1	aws-bluage-runtime-fargate-dev-343763094578-us-west-1
USA West (Oregon)	aws-bluage-runtime-fargate-rel-688933007849-us-west-2	aws-bluage-runtime-fargate-dev-688933007849-us-west-2
Europa (Irland)	aws-bluage-runtime-fargate-rel-140138033705-eu-west-1	aws-bluage-runtime-fargate-dev-140138033705-eu-west-1
Europa (Paris)	aws-bluage-runtime-fargate-rel-339712948211-eu-west-3	aws-bluage-runtime-fargate-dev-339712948211-eu-west-3
Europa (Frankfurt)	aws-bluage-runtime-fargate-rel-339712918892-eu-central-1	aws-bluage-runtime-fargate-dev-339712918892-eu-central-1
Südamerika (São Paulo)	aws-bluage-runtime-fargate-rel-767397998881-sa-east-1	aws-bluage-runtime-fargate-dev-767397998881-sa-east-1
Asien-Pazifik (Tokio)	aws-bluage-runtime-fargate-rel-891377400849-ap-northeast-1	aws-bluage-runtime-fargate-dev-891377400849-ap-northeast-1

AWS-Region	Bucket veröffentlichen	Bucket vor der Veröffentlichung
Asien-Pazifik (Sydney)	aws-bluage-runtime-fargate-rel-533267435478-ap-southeast-2	aws-bluage-runtime-fargate-dev-533267435478-ap-southeast-2

## Verwenden von, um den Inhalt des Buckets aufzulisten AWS CLI

Nach dem Onboarding können Sie den Inhalt des Buckets auflisten, indem Sie den folgenden AWS CLI Befehl in einem Terminal ausführen.

```
aws s3 ls bucket-name
```

*bucket-name* Ersetzen Sie ihn durch den Namen des Buckets für Ihre AWS-Region aus der vorherigen Tabelle.

Dieser Befehl gibt eine Liste von Ordnern zurück, die verschiedenen Versionen der (nicht verwalteten) AWS Blu Age Runtime-Laufzeit entsprechen, z. B. die folgende für einen Release-Bucket:

```
PRE 3.10.0/  
PRE 4.0.0/
```

Oder wie folgt für einen Build-Bucket:

```
PRE 4.1.0-alpha.8/  
PRE 4.1.0-alpha.9/
```

Wir empfehlen, dass Sie die neueste verfügbare Version verwenden. Wenn das nicht möglich ist, verwenden Sie die Runtime-Version, die während der Refactoring-Phase der Anwendung validiert wurde. Führen Sie den folgenden Befehl aus, um die verfügbaren Frameworks für eine bestimmte Version aufzulisten:

```
aws s3 ls s3://bucket-name/version/Framework/
```

*bucket-name* Ersetzen Sie es durch den Namen des Buckets für Ihre AWS-Region und *version* durch die gewünschte Version. Im Folgenden finden Sie zwei Beispiele.

Für einen Release-Bucket:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/  
Framework/
```

Der Befehl gibt eine Liste von Frameworks zurück, wie zum Beispiel:

```
2024-04-08 16:11:19 152040176 aws-bluage-runtime-4.0.0.tar.gz  
2024-04-08 16:11:50          45 aws-bluage-runtime-4.0.0.tar.gz.checksumSHA256  
2024-04-08 16:11:52 176518889 aws-bluage-webapps-4.0.0.tar.gz  
2024-04-08 16:12:28          45 aws-bluage-webapps-4.0.0.tar.gz.checksumSHA256
```

Für einen Build-Bucket:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-dev-139023371234-us-  
east-1/4.1.0-alpha.9/Framework/
```

Der Befehl gibt eine Liste von Frameworks zurück, wie zum Beispiel:

```
2024-04-09 20:23:34 152304534 aws-bluage-runtime-4.1.0-alpha.9.tar.gz  
2024-04-09 20:24:05          45 aws-bluage-runtime-4.1.0-alpha.9.tar.gz.checksumSHA256  
2024-04-09 20:24:07 176262381 aws-bluage-webapps-4.1.0-alpha.9.tar.gz  
2024-04-09 20:24:42          45 aws-bluage-webapps-4.1.0-alpha.9.tar.gz.checksumSHA256
```

## Laden Sie das Framework herunter

Sie können das Framework herunterladen, um beispielsweise die AWS Blu Age Runtime-Version auf einer vorhandenen Amazon EC2 EC2-Instance zu aktualisieren.

```
aws s3 cp s3://bucket-name/version/Framework/ folder-of-your-choice --  
recursive
```

Wobei gilt:

Ordner Ihrer Wahl

Ordnerpfad, in den Sie das Framework herunterladen möchten.

```
Beispiel: aws s3 cp s3://aws-bluage-runtime-artifacts-139023371234-us-  
east-1/4.0.0/Framework/ . --recursive
```

Die Ausgabe dieses Befehls sieht wie folgt aus.

```
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/
Framework/aws-bluage-runtime-4.0.0.tar.gz.checksumSHA256 to ./aws-bluage-
runtime-4.0.0.tar.gz.checksumSHA256
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/
Framework/aws-bluage-webapps-4.0.0.tar.gz.checksumSHA256 to ./aws-bluage-
webapps-4.0.0.tar.gz.checksumSHA256
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/Framework/aws-
bluage-webapps-4.0.0.tar.gz to ./aws-bluage-webapps-4.0.0.tar.gz
download: s3://aws-bluage-runtime-artifacts-139023371234-us-east-1/4.0.0/Framework/aws-
bluage-runtime-4.0.0.tar.gz to ./aws-bluage-runtime-4.0.0.tar.gz
```

Sie können die Framework-Dateien wie folgt auflisten:

```
ls -l
```

Die Ausgabe dieses Befehls sieht wie folgt aus.

```
total 230928
-rw-rw-r-- 1 cloudshell-user cloudshell-user 152040176 Apr  8 16:11 aws-bluage-
runtime-4.0.0.tar.gz
-rw-rw-r-- 1 cloudshell-user cloudshell-user          45 Apr  8 16:11 aws-bluage-
runtime-4.0.0.tar.gz.checksumSHA256
-rw-rw-r-- 1 cloudshell-user cloudshell-user 176518889 Apr  8 16:11 aws-bluage-
webapps-4.0.0.tar.gz
-rw-rw-r-- 1 cloudshell-user cloudshell-user          45 Apr  8 16:12 aws-bluage-
webapps-4.0.0.tar.gz.checksumSHA256
```

## Anforderungen an die Einrichtung der Infrastruktur für AWS Blu Age Runtime (nicht verwaltet)

In diesem Thema wird die minimale Infrastrukturkonfiguration beschrieben, die für die Ausführung von AWS Blu Age Runtime (nicht verwaltet) erforderlich ist. In den folgenden Verfahren wird beschrieben, wie Sie AWS Blu Age Runtime (nicht verwaltet) auf dem Computer Ihrer Wahl einrichten, um eine modernisierte Anwendung auf der AWS Blu Age Runtime bereitzustellen. Die Ressourcen, die Sie erstellen, müssen sich in einer Amazon-VPC VPC, die über ein Subnetz verfügt, das Ihrer Anwendungsdomäne zugewiesen ist.

### Themen

- [Anforderungen an die Infrastruktur](#)
- [Amazon EC2-Instance-Typen für AWS Blu Age Runtime \(auf Amazon EC2\)](#)

- [AWS Blu Age Runtime auf Amazon EC2 ausführen](#)
- [AWS Blu Age Runtime auf Amazon ECS auf Amazon EC2 ausführen](#)
- [AWS Blu Age Runtime auf Amazon EKS auf Amazon EC2 ausführen](#)
- [Ausführen von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)

## Anforderungen an die Infrastruktur

### Eine Sicherheitsgruppe erstellen

Wenn Sie planen, an Amazon EC2 EC2-Instances auf Amazon EKS zu arbeiten, überspringen Sie dieses Verfahren, da der Amazon EKS-Clustererstellungsprozess in Ihrem Namen eine Sicherheitsgruppe erstellt. Verwenden Sie diese Sicherheitsgruppe in den folgenden Verfahren, anstatt eine neue zu erstellen.

1. Öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im linken Navigationsbereich unter Sicherheit die Option Sicherheitsgruppen aus.
3. Wählen Sie im mittleren Bereich die Option Sicherheitsgruppe erstellen aus.
4. Geben Sie in das Feld Name der Sicherheitsgruppe ein **M2BluagePrivateLink-SG**.
5. Wählen Sie im Abschnitt Eingehende Regeln die Option Regel hinzufügen aus.
6. Wählen Sie als Typ die Option HTTPS aus.
7. Geben Sie als Quelle Ihre VPC CIDR ein.
8. Wählen Sie im Abschnitt Regeln für ausgehende Nachrichten die Option Regel hinzufügen aus.
9. Wählen Sie als Typ die Option HTTPS aus.
10. Geben Sie für Destination **0.0.0.0/0** ein.
11. Wählen Sie Sicherheitsgruppe erstellen aus.

### Erstellen Sie einen Amazon VPC-Endpunkt

1. Öffnen Sie die Amazon VPC-Konsole unter <https://console.aws.amazon.com/vpc/>.
2. Wählen Sie im linken Navigationsbereich unter Virtual Private Cloud die Option Endpoints aus.
3. Wählen Sie im mittleren Bereich die Option Endpunkt erstellen aus.
4. Geben Sie im Abschnitt Services **SQS** in das Suchfeld ein und wählen Sie dann den Amazon SQS-Service aus, der Ihrer Region entspricht.
5. Wählen Sie im Abschnitt VPC die Amazon-VPC aus, die Sie im vorherigen Schritt erstellt haben.

6. Wählen Sie im Abschnitt Subnetze das Subnetz aus, das Sie für Ihre Anwendungsdomäne erstellt haben.
7. Wählen Sie im Abschnitt Sicherheitsgruppen die Sicherheitsgruppe aus dem vorherigen Verfahren aus.
8. Wählen Sie Endpunkt erstellen aus.

### Eine IAM-Richtlinie erstellen

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich unter Zugriffsverwaltung die Option Richtlinien aus.
3. Wählen Sie im mittleren Bereich die Option Richtlinie erstellen aus.
4. Wählen Sie im Abschnitt Richtlinien-Editor JSON aus.
5. Ersetzen Sie alle JSON-Dateien, die Sie im Editor sehen, durch die folgende JSON-Datei.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "sqs:GetQueueUrl",
        "sqs:ReceiveMessage",
        "sqs:SendMessage"
      ],
      "Resource": "*"
    }
  ]
}
```

#### Note

Wenn Sie weitere Informationen zur Anpassung Ihrer Policy benötigen, wenden Sie sich an Ihren AWS Blu Age-Liefermanager oder Kundenbetreuer.

6. Wählen Sie Weiter aus.
7. Geben Sie einen Namen für die Richtlinie ein und wählen Sie dann Richtlinie erstellen aus.



## Erstellen einer IAM-Rolle

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im linken Navigationsbereich unter Zugriffsverwaltung die Option Rollen aus.
3. Wählen Sie im mittleren Bereich die Option Rolle erstellen aus.
4. Wählen Sie im Abschnitt Anwendungsfall je nach Ihrer Computerauswahl eine der folgenden Optionen aus:
  - EC2 (für Amazon EC2 und Amazon EKS auf Amazon EC2)
  - Elastic Container Service und dann EC2-Rolle für Elastic Container Service (für Amazon ECS auf Amazon EC2)
  - Elastic Container Service und dann Elastic Container Service Task (für Amazon ECS, verwaltet von Fargate)
5. Wählen Sie Weiter aus.
6. Geben Sie im Suchfeld den Namen der Richtlinie ein, die Sie zuvor erstellt haben.
7. Wählen Sie das Kontrollkästchen links neben Ihrer Richtlinie aus.

### Note

Wenn Sie keine Richtlinie hinzufügen können, beenden Sie die Erstellung der Rolle und aktualisieren Sie dann die Rolle, um die Richtlinie hinzuzufügen.

8. Wählen Sie Weiter aus.
9. Geben Sie einen Namen für die Rolle ein, wählen Sie dann Rolle erstellen aus.

## Amazon EC2-Instance-Typen für AWS Blu Age Runtime (auf Amazon EC2)

Im Folgenden finden Sie eine Liste der Amazon EC2-Instance-Typen, die Sie für AWS Blu Age Runtime (auf Amazon EC2) verwenden können, wenn Sie Amazon EC2 EC2-Instances erstellen oder Amazon EKS-Worker-Knoten definieren.

```
t3.xlarge
t3.small
t3.large
t2.small
t2.large
r7a.medium
```

r7a.large  
r7a.xlarge  
r7a.2xlarge  
r7a.4xlarge  
r7a.8xlarge  
r7a.12xlarge  
r7a.16xlarge  
r7a.24xlarge  
r7a.32xlarge  
r7a.48xlarge  
r7a.metal-48xl  
r7i.large  
r7i.xlarge  
r7i.2xlarge  
r7i.4xlarge  
r7i.8xlarge  
r7i.12xlarge  
r7i.16xlarge  
r7i.24xlarge  
r7i.48xlarge  
r7i.metal-24xl  
r7i.metal-48xl  
r6i.xlarge  
r6i.large  
r6i.4xlarge  
r6i.2xlarge  
r5b.xlarge  
r5b.large  
r5b.2xlarge  
r3.xlarge  
m6i.xlarge  
m6i.large  
m6i.8xlarge  
m6i.4xlarge  
m6i.2xlarge  
m6i.16xlarge  
m5zn.xlarge  
m5zn.large  
m5zn.3xlarge  
m5zn.2xlarge  
m5.xlarge  
m5.large  
m5.8xlarge  
m5.4xlarge

```
m5.2xlarge
m5.16xlarge
m5.12xlarge
c6i.xlarge
c6i.large
c6i.8xlarge
c6i.4xlarge
c6i.2xlarge
c6i.16xlarge
c5.xlarge
c5.large
c5.9xlarge
c5.4xlarge
c5.2xlarge
c5.18xlarge
c5.12xlarge
```

## AWS Blu Age Runtime auf Amazon EC2 ausführen

Gehen Sie wie folgt vor, um eine Amazon EC2 EC2-Instance zu erstellen.

### Erstellen einer Amazon-EC2-Instance

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie Launch Instance (Instance starten) aus.
3. Wählen Sie unter Instance-Typ einen der unter aufgeführten Typen aus. [the section called “Amazon EC2-Instance-Typen für AWS Blu Age Runtime \(auf Amazon EC2\)”](#)
4. Wählen Sie im Abschnitt key pair entweder ein vorhandenes Schlüsselpaar aus oder erstellen Sie ein neues.
5. Wählen Sie im Abschnitt Netzwerkeinstellungen die Option Bestehende Sicherheitsgruppe auswählen aus.
6. Wählen Sie für Allgemeine Sicherheitsgruppen M2 BluagePrivate Link-SG aus.
7. Erweitern Sie den Abschnitt Erweiterte Details.
8. Wählen Sie für das IAM-Instanzprofil die IAM-Rolle aus, die Sie zuvor erstellt haben.
9. Wählen Sie Launch Instance (Instance starten) aus.

## Installieren Sie die Anwendung auf der Amazon EC2 EC2-Instance

1. Wenn sich der Status der Amazon EC2 EC2-Instance auf Running ändert, stellen Sie eine Verbindung mit der Instance her.
2. Installieren Sie die folgenden Softwarekomponenten auf der Instance:
  - Java Runtime Environment (JRE) 17.
  - Apache Tomcat 10.
  - AWS Blu Age Runtime (auf Amazon EC2). Installieren Sie die AWS Blu Age Runtime im Stammverzeichnis des Apache Tomcat-Installationsordners (einige Dateien werden hinzugefügt, während andere überschrieben werden).

Um die zusätzlichen Webapps zu installieren, die zusammen mit dem AWS Blu Age Runtime-Archiv geliefert werden, richten Sie eine sekundäre Instanz des Apache Tomcat-Servers ein und dekomprimieren Sie das Webapps-Archiv an diesem Speicherort.

## AWS Blu Age Runtime auf Amazon ECS auf Amazon EC2 ausführen

1. Erstellen Sie einen Amazon ECS-Cluster mit Amazon EC2 EC2-Instances als zugrunde liegende Infrastruktur. Weitere Informationen finden Sie unter [Erste Schritte mit Windows auf Amazon EC2](#) im Amazon Elastic Container Service Developer Guide.
2. Geben Sie die IAM-Rolle an, die Sie in den vorherigen Schritten erstellt haben.
3. Wählen Sie einen der Instanztypen aus, die unter aufgeführt sind. [the section called "Amazon EC2-Instance-Typen für AWS Blu Age Runtime \(auf Amazon EC2\)"](#)
4. Wählen Sie unter Netzwerkeinstellungen für Amazon EC2 EC2-Instances die Sicherheitsgruppe aus, die Sie in den vorherigen Schritten erstellt haben.

## AWS Blu Age Runtime auf Amazon EKS auf Amazon EC2 ausführen

1. Erstellen Sie einen Amazon-EKS-Cluster. Weitere Informationen finden Sie unter [Erstellen eines Amazon EKS-Clusters](#) im Amazon EKS-Benutzerhandbuch.
2. Wie bereits erwähnt, wird in Ihrem Namen eine Sicherheitsgruppe erstellt. Sie können diese Sicherheitsgruppe verwenden, wenn Sie den Amazon VPC-Endpoint erstellen.
3. Erstellen Sie eine Knotengruppe. Geben Sie die IAM-Rolle an, die Sie in den vorherigen Schritten erstellt haben.

4. Wählen Sie einen der Instanztypen aus, die unter aufgeführt sind. [the section called “Amazon EC2-Instance-Typen für AWS Blu Age Runtime \(auf Amazon EC2\)”](#)
5. Amazon EKS weist die Sicherheitsgruppe automatisch den gestarteten Amazon EC2 EC2-Instances zu.

## Ausführen von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

Erstellen Sie einen Amazon ECS-Cluster mit AWS Fargate (serverlos) als zugrunde liegende Infrastruktur. Weitere Informationen finden Sie unter Erste Schritte mit Linux-Containern auf AWS Fargate im Amazon Elastic Container Service Developer Guide.

## AWS Bereitstellung von Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

Die Themen in diesem Abschnitt beschreiben, wie Sie AWS Blu Age Runtime auf Amazon ECS Managed von einrichten AWS Fargate, wie Sie die Runtime-Version aktualisieren, wie Sie Ihre Bereitstellung mithilfe von Amazon CloudWatch Alarms überwachen und wie Sie lizenzierte Abhängigkeiten hinzufügen.

### Themen

- [Einrichtung von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)
- [Aktualisierung der AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)
- [Amazon CloudWatch Alarms for AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)
- [Einrichtung lizenzierter Abhängigkeiten in AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate](#)

## Einrichtung von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

In diesem Thema wird erklärt, wie Sie die PlanetsDemo Beispielanwendung mithilfe von AWS Blu Age Runtime auf Amazon ECS einrichten und bereitstellen, verwaltet von AWS Fargate.

AWS Blu Age Runtime auf Amazon ECS, verwaltet von, AWS Fargate ist für Linux/X86 verfügbar.

### Themen

- [Voraussetzungen](#)

- [Einrichtung](#)
- [Testen Sie die bereitgestellte Anwendung](#)

## Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- Konfigurieren Sie das, AWS CLI indem Sie den Schritten [unter Konfiguration der AWS-CLI](#) folgen.
- [the section called "AWS Voraussetzungen für die Laufzeit von Blu Age"](#)Vollständig und [the section called "AWS Einführung in Blu Age Runtime"](#).
- Laden Sie die AWS Blu Age Runtime auf Amazon ECS herunter, die von AWS Fargate Binärdateien verwaltet wird. Anweisungen finden Sie unter [the section called "AWS Einführung in Blu Age Runtime"](#).
- Laden Sie die Apache Tomcat 10-Binärdateien herunter.
- Laden Sie das [PlanetsDemoAnwendungsarchiv](#) herunter.
- Erstellen Sie eine Amazon Aurora PostgreSQL-Datenbank für JICS und führen Sie die `PlanetsDemo-v1/jics/sql/initJics.sql` Abfrage darauf aus. Informationen zum Erstellen einer Amazon Aurora PostgreSQL-Datenbank finden Sie unter [Erstellen und Herstellen einer Verbindung zu einem Aurora PostgreSQL-DB-Cluster](#).

## Einrichtung

Gehen Sie wie folgt vor, um die PlanetsDemo Beispielanwendung einzurichten.


1. Extrahieren Sie nach dem Herunterladen der Apache Tomcat-Binärdateien den Inhalt und wechseln Sie in den `conf` Ordner. Öffnen Sie die `catalina.properties` Datei zur Bearbeitung und ersetzen Sie die Zeile, die `common.loader` mit der folgenden Zeile beginnt.

```
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. Komprimieren Sie den Apache Tomcat-Ordner, indem Sie den Befehl `tar` verwenden, um ein ``tar.gz``-Archiv zu erstellen.
3. Bereiten Sie ein [Dockerfile](#) vor, um Ihr benutzerdefiniertes Image auf der Grundlage der bereitgestellten Runtime-Binärdateien und der Apache Tomcat-Server-Binärdateien zu erstellen.

Sehen Sie sich das folgende Dockerfile-Beispiel an. Das Ziel besteht darin, Apache Tomcat 10 zu installieren, gefolgt von AWS Blu Age Runtime (für Amazon ECS, verwaltet von AWS Fargate), die im Stammverzeichnis des Apache Tomcat 10-Installationsverzeichnisses extrahiert wurde, und dann die modernisierte Beispielanwendung mit dem Namen zu installieren.

PlanetsDemo

 Note

Der Inhalt der Skripten `install-gapwalk.sh` und `install-app.sh`, die in diesem Dockerfile-Beispiel verwendet werden, wird nach dem Dockerfile aufgeführt.

```
FROM --platform=linux/x86_64 amazonlinux:2

RUN mkdir -p /workdir/apps
WORKDIR /workdir
COPY install-gapwalk.sh .
COPY install-app.sh .
RUN chmod +x install-gapwalk.sh
RUN chmod +x install-app.sh

# Install Java and AWS CLI v2-y
RUN yum install sudo java-17-amazon-corretto unzip tar -y
RUN sudo yum remove awscli -y
RUN curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
  "awscliv2.zip"
RUN sudo unzip awscliv2.zip
RUN sudo ./aws/install

# Installation dir
RUN mkdir -p /usr/local/velocity/installation/gapwalk
# Copy PlanetsDemo archive to a dedicated apps dir
COPY PlanetsDemo-v1.zip /workdir/apps/

# Copy resources (tomcat, blu age runtime) to installation dir
COPY tomcat.tar.gz /usr/local/velocity/installation/tomcat.tar.gz
COPY aws-bluage-on-fargate-runtime-4.x.x.tar.gz /usr/local/velocity/installation/
gapwalk/gapwalk-bluage-on-fargate.tar.gz

# run relevant installation scripts
RUN ./install-gapwalk.sh
```

```
RUN ./install-app.sh

EXPOSE 8080
EXPOSE 8081
# ...

# Run Command to start Tomcat server
CMD ["sh", "-c", "sudo /bluage-on-fargate/tomcat.gapwalk/velocity/startup.sh
  $ECS_CONTAINER_METADATA_URI_V4 $AWS_CONTAINER_CREDENTIALS_RELATIVE_URI"]
```

Im Folgenden finden Sie den Inhalt von `install-gapwalk.sh`.

```
#!/bin/sh

# Vars
TEMP_DIR=/bluage-on-fargate/tomcat.gapwalk/temp

# Install
echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage-on-fargate
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk-bluage-on-fargate.tar.gz
  ${TEMP_DIR}
sudo cp /usr/local/velocity/installation/tomcat.tar.gz ${TEMP_DIR}
# Create velocity dir
mkdir -p /bluage-on-fargate/tomcat.gapwalk/velocity
# Extract tomcat files
tar -xvf ${TEMP_DIR}/tomcat.tar.gz -C ${TEMP_DIR}
# Copy all tomcat files to velocity dir
cp -fr ${TEMP_DIR}/apache-tomcat-10.x.x/* /bluage-on-fargate/tomcat.gapwalk/
velocity
# Remove default webapps of Tomcat
rm -f /bluage-on-fargate/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk-bluage-on-fargate.tar.gz -C /bluage-on-fargate/
tomcat.gapwalk
# Remove temporary extraction dir
sudo rm -rf ${TEMP_DIR}
```

Im Folgenden sind die Inhalte von `install-app.sh` aufgeführt.



```
#!/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. Geben Sie die Verbindungsinformationen für die Datenbank, die Sie als Teil der Voraussetzungen im folgenden Codeausschnitt erstellt haben, in die `application-main.yml` Datei ein, die sich im Ordner befindet. `{TOMCAT_GAPWALK_DIR}/config` Weitere Informationen finden Sie unter [Erstellen eines Aurora PostgreSQL-DB-Clusters und Herstellen einer Verbindung zu diesem](#).

```
datasource:
  jicsDs:
    driver-class-name :
    url:
    username:
    password:
    type :
```

5. Erstellen Sie das Image und übertragen Sie es in Ihr Amazon ECR-Repository. Anweisungen finden Sie unter [Pushing a Docker-Image](#) im Amazon Elastic Container Registry-Benutzerhandbuch.
6. Öffnen Sie die Konsole unter <https://console.aws.amazon.com/ecs/v2>.
7. Wählen Sie im linken Navigationsbereich Aufgabendefinitionen aus.
8. Wählen Sie als Starttyp die Option AWS Fargate.
9. Wählen Sie die Aufgabenrolle aus, die Sie als Teil von erstellt haben [the section called "Anforderungen an die Einrichtung der Infrastruktur"](#).
10. Hängen Sie Ihr Bild an den Container an.
11. Füllen Sie das Formular vollständig aus und wählen Sie dann Erstellen.
12. Wählen Sie im linken Navigationsbereich Clusters und anschließend Ihren Cluster aus der Liste aus.
13. Wählen Sie auf der Detailseite Ihres Clusters auf der Registerkarte Services die Option Create aus.

14. Wählen Sie die Aufgabendefinition aus.
15. Erweitern Sie den Abschnitt Netzwerk und konfigurieren Sie die VPC, die Subnetze und die Sicherheitsgruppe, die Sie als Teil erstellt haben. [the section called “Anforderungen an die Einrichtung der Infrastruktur”](#)
16. Stellen Sie Ihren Amazon ECS-Service bereit.

Wenn die Bereitstellung fehlschlägt, überprüfen Sie die Protokolle. Um sie zu finden, gehen Sie auf die Aufgabenseite in Amazon ECS, verwaltet von AWS Fargate, und wählen Sie dann die Registerkarte Protokolle. Wenn Sie Fehlercodes finden, die mit einem C gefolgt von einer Zahl beginnen, z. B. CXXXX, notieren Sie sich die Fehlermeldungen. Beispielsweise ist der Fehlercode C5102 ein häufiger Fehler, der auf eine falsche Infrastrukturkonfiguration hinweist. Sie können auch innerhalb Ihrer laufenden Aufgabe navigieren und einige Befehle ausführen, ähnlich wie bei AWS Blue Age Runtime (auf Amazon EC2). Weitere Informationen finden Sie unter [Verwenden von Amazon ECS Exec für das Debuggen](#) im Amazon Elastic Container Service Developer Guide.

Um eine interaktive Shell zu öffnen, führen Sie den folgenden Befehl auf Ihrem lokalen Computer aus.

```
aws ecs execute-command --cluster your_cluster_name --container your_container_name --  
task task_id --interactive --command /bin/sh
```

Testen Sie die bereitgestellte Anwendung

Ein Beispiel für das Testen der PlanetsDemo Anwendung finden Sie unter [the section called “Testen Sie die Anwendung PlanetsDemo”](#).

## Aktualisierung der AWS Blue Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

In diesem Handbuch wird beschrieben, wie Sie die AWS Blue Age Runtime auf Amazon ECS aktualisieren, verwaltet von AWS Fargate.

Themen

- [Voraussetzungen](#)
- [Aktualisieren Sie die AWS Blue Age Runtime](#)

## Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Laden Sie die Version der AWS Blu Age Runtime herunter, auf die Sie ein Upgrade durchführen möchten. Weitere Informationen finden Sie unter [the section called “AWS Einführung in Blu Age Runtime”](#). Das Framework besteht aus zwei Binärdateien: `aws-bluage-runtime-x.x.x.x.tar.gz` und `aws-bluage-webapps-x.x.x.x.tar.gz`.

## Aktualisieren Sie die AWS Blu Age Runtime

Gehen Sie wie folgt vor, um die AWS Blu Age Runtime zu aktualisieren.

1. Erstellen Sie Ihr Docker-Image mit der gewünschten AWS Blu Age Runtime-Version neu. Anweisungen finden Sie unter [the section called “Einrichtung von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate”](#).
2. Übertragen Sie Ihr Docker-Image in Ihr Amazon ECR-Repository.
3. Beenden Sie Ihren Amazon ECS-Service und starten Sie ihn neu.
4. Überprüfen Sie die Protokolle.

Die AWS Blu Age Runtime wurde erfolgreich aktualisiert.

## Amazon CloudWatch Alarms for AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

Um besser sichtbare Benachrichtigungen zu erhalten, wenn Ihre bereitgestellten Anwendungen auf Ausnahmen stoßen, richten Sie CloudWatch den Empfang Ihres Anwendungsprotokolls ein und fügen Sie einen Alarm hinzu, der Sie vor möglichen Fehlern warnt.

### Einrichtung des Alarms

Mithilfe von CloudWatch Protokollen können Sie je nach Ihrer Anwendung und Ihren Anforderungen eine beliebige Anzahl von Metriken und Alarmen konfigurieren.

Insbesondere können Sie direkt während der Erstellung Ihres Amazon ECS-Clusters proaktive Alarme für Nutzungswarnungen einrichten, sodass Sie benachrichtigt werden, wenn Fehler auftreten. Um Fehler bei der Verbindung zum AWS Blu Age-Kontrollsystem hervorzuheben, fügen Sie in den

Protokollen eine Metrik hinzu, die sich auf die Zeichenfolge „Fehler C“ bezieht. Anschließend können Sie einen Alarm definieren, der auf diese Metrik reagiert.

## Einrichtung lizenzierter Abhängigkeiten in AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

In diesem Thema wird beschrieben, wie Sie zusätzliche lizenzierte Abhängigkeiten einrichten, die Sie mit AWS Blu Age Runtime auf Amazon ECS, verwaltet von, verwenden können AWS Fargate.

### Themen

- [Voraussetzungen](#)
- [Übersicht](#)

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) Vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Holen Sie sich die folgenden Abhängigkeiten aus ihrer Quelle.

### Oracle-Datenbank

Stellen Sie einen [Oracle-Datenbanktreiber bereit](#). Zum Beispiel ojdbc11-23.3.0.23.09.jar.

### IBM MQ-Verbindung

Stellen Sie einen [IBM MQ-Client bereit](#). Zum Beispiel com.ibm.mq.jakarta.client-9.3.4.1.jar.

Geben Sie mit dieser Abhängigkeitsversion auch die folgenden transitiven Abhängigkeiten an:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15to18-1.76.jar
- bcutil-jdk15to18-1.76.jar

### DDS-Druckerdateien

Stellen Sie die [Jasper-Berichtsbibliothek bereit](#). Zum Beispiel jasperreports-6.16.0.jar, aber eine neuere Version könnte kompatibel sein.

Geben Sie mit dieser Abhängigkeitsversion auch die folgenden transitiven Abhängigkeiten an:

- 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-sammlungen-3.2.2.jar

## Übersicht

Gehen Sie wie folgt vor, um die Abhängigkeiten zu installieren.

1. Kopieren Sie nach Bedarf eine der oben genannten Abhängigkeiten in Ihren Docker-Image-Build-Ordner.
2. Wenn Ihre JICS- oder Blusam-Datenbank auf Oracle gehostet wird, stellen Sie den Oracle-Datenbanktreiber unter bereit. *your-tomcat-path/extra*
3. Kopieren Sie diese Abhängigkeiten auf Ihrem Dockerfile nach. *your-tomcat-path/extra*
4. Erstellen Sie Ihr Docker-Image und übertragen Sie es dann auf Amazon ECR.
5. Beenden Sie Ihren Amazon ECS-Service und starten Sie ihn neu.
6. Überprüfen Sie die Protokolle.

## AWS Bereitstellung von Blu Age Runtime auf Amazon EC2

Die Themen in diesem Abschnitt beschreiben, wie Sie AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2 einrichten, wie Sie die Runtime-Version aktualisieren, wie Sie Ihre Bereitstellung mithilfe von CloudWatch Amazon-Alarmen überwachen und wie Sie lizenzierte Abhängigkeiten hinzufügen. Diese Anweisungen gelten sowohl für die Erstellung von Amazon EC2-Instances als auch für die Verwendung von Amazon ECS auf Amazon EC2 oder Amazon EKS auf Amazon EC2.

## Themen

- [Einrichtung von AWS Blu Age Runtime \(nicht verwaltet\) auf Amazon EC2](#)
- [Verwenden von Containern in Amazon EC2 für Amazon ECS und Amazon EKS](#)
- [Aktualisierung der AWS Blu Age Runtime auf Amazon EC2](#)
- [AWS Blu Age Runtime \(auf Amazon EC2\) Amazon Alarms CloudWatch](#)
- [Einrichten lizenzierter Abhängigkeiten in AWS Blu Age Runtime auf Amazon EC2](#)

## Einrichtung von AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2

In diesem Thema wird erklärt, wie Sie die PlanetsDemo Beispielanwendung mithilfe von AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2 einrichten und bereitstellen.

### Themen

- [Voraussetzungen](#)
- [Einrichtung](#)
- [Testen Sie die bereitgestellte Anwendung](#)

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- Konfigurieren Sie das, AWS CLI indem Sie den Schritten [unter Konfiguration der AWS-CLI](#) folgen.
- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) Vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Erstellen Sie eine Amazon EC2 EC2-Instance mit einem der unterstützten Instance-Typen. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon-EC2-Linux-Instancen](#).
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur Amazon EC2 EC2-Instance herstellen können, z. B. mithilfe von SSM.
- Laden Sie AWS Blu Age Runtime (auf Amazon EC2) unter *your-tomcat-path*/\* herunter und entpacken Sie es. Stellen Sie sicher, dass Sie die `bluage.bin` Datei genau an dem Ort platzieren, der in der `CATALINA_HOME` Umgebungsvariablen angegeben ist, die in der Apache Tomcat-Dokumentation unter [CATALINA\\_HOME und CATALINA\\_BASE](#) beschrieben ist. Anweisungen zum Abrufen der Blu Age Runtime finden Sie unter. AWS [the section called “AWS Einführung in Blu Age Runtime”](#)

- Laden Sie das [PlanetsDemoAnwendungsarchiv](#) herunter.
- Entpacken Sie das Archiv und laden Sie die Anwendung in einen Amazon S3 S3-Bucket Ihrer Wahl hoch.
- Erstellen Sie eine Amazon Aurora PostgreSQL-Datenbank für JICS und führen Sie die `PlanetsDemo-v1/jics/sql/initJics.sql` Abfrage darauf aus. Informationen zum Erstellen einer Amazon Aurora PostgreSQL-Datenbank finden Sie unter [Erstellen und Herstellen einer Verbindung zu einem Aurora PostgreSQL-DB-Cluster](#).

## Einrichtung

Gehen Sie wie folgt vor, um die PlanetsDemo Beispielanwendung einzurichten.

1. Connect zu Ihrer Amazon EC2 EC2-Instance her und wechseln Sie zu dem `conf` Ordner unter Ihrem Apache Tomcat 10-Installationsordner. Öffnen Sie die `catalina.properties` Datei zur Bearbeitung und ersetzen Sie die Zeile, die `common.loader` mit `beginnt`, durch die folgende Zeile.

```
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. Navigieren Sie zum Verzeichnis `<your-tomcat-path>/webapps`.
3. Kopieren Sie mit dem PlanetsDemo folgenden Befehl die im Ordner `PlanetsDemo -v1/webapps/` verfügbaren Binärdateien aus dem Amazon S3 S3-Bucket.

```
aws s3 cp s3://path-to-demo-app-webapps/ . --recursive
```

### Note

`path-to-demo-app-webapps` Ersetzen Sie es durch die richtige Amazon S3 S3-URI für den Bucket, in den Sie das PlanetsDemo Archiv zuvor entpackt haben.

4. Kopieren Sie den Inhalt des `PlanetsDemo-v1/config/` Ordners in. `<your-tomcat-path>/config/`
5. Geben Sie die Verbindungsinformationen für die Datenbank an, die Sie als Teil der Voraussetzungen im folgenden Codeausschnitt in der Datei erstellt haben. `application-`

`main.yml` Weitere Informationen finden Sie unter [Erstellen eines Aurora PostgreSQL-DB-Clusters und Herstellen einer Verbindung zu diesem](#).

```
datasource:
  jicsDs:
    driver-class-name :
    url:
    username:
    password:
    type :
```

6. Starten Sie Ihren Apache Tomcat-Server und überprüfen Sie die Protokolle.

```
your-tomcat-path/startup.sh

tail -f your-tomcat-path/logs/catalina.log
```

Wenn Sie Fehlercodes finden, die mit einem C gefolgt von einer Zahl beginnen, z. B. CXXXX, notieren Sie sich die Fehlermeldungen. Beispielsweise ist der Fehlercode C5102 ein häufiger Fehler, der auf eine falsche Infrastrukturkonfiguration hinweist.

Testen Sie die bereitgestellte Anwendung

Ein Beispiel für das Testen der PlanetsDemo Anwendung finden Sie unter [the section called “Testen Sie die Anwendung PlanetsDemo”](#).

## Verwenden von Containern in Amazon EC2 für Amazon ECS und Amazon EKS

In diesem Thema wird erklärt, wie Sie die PlanetsDemo Beispielanwendung mithilfe von AWS Blueprints (nicht verwaltet) auf Amazon EC2 als Container einrichten und bereitstellen.

Themen

- [Voraussetzungen](#)
- [Einrichtung](#)
- [Testen Sie die bereitgestellte Anwendung](#)

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.



- Konfigurieren Sie das, AWS CLI indem Sie den Schritten [unter Konfiguration der AWS-CLI](#) folgen.
- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) Vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Laden Sie AWS Blu Age Runtime (auf Amazon EC2) herunter. Anweisungen zum Abrufen der Runtime finden Sie unter [the section called “AWS Einführung in Blu Age Runtime”](#).
- Laden Sie das [PlanetsDemoAnwendungsarchiv](#) herunter.
- Erstellen Sie eine Amazon Aurora PostgreSQL-Datenbank für JICS und führen Sie die `PlanetsDemo-v1/jics/sql/initJics.sql` Abfrage darauf aus. Informationen zum Erstellen einer Amazon Aurora PostgreSQL-Datenbank finden Sie unter [Erstellen und Herstellen einer Verbindung zu einem Aurora PostgreSQL-DB-Cluster](#).

## Einrichtung

Gehen Sie wie folgt vor, um die PlanetsDemo Beispielanwendung einzurichten.

1. Bereiten Sie ein [Dockerfile](#) vor, um Ihr benutzerdefiniertes Image auf der Grundlage der bereitgestellten Runtime-Binärdateien und der Apache Tomcat-Server-Binärdateien zu erstellen. Sehen Sie sich das folgende Dockerfile-Beispiel an. Ziel ist es, Apache Tomcat 10 zu installieren, gefolgt von AWS Blu Age Runtime (auf Amazon EC2), die im Stammverzeichnis des Apache Tomcat 10-Installationsverzeichnisses extrahiert wurde, und dann die modernisierte Beispielanwendung mit dem Namen zu installieren. PlanetsDemo Die `install-app.sh` Skripte `install-gapwalk.sh` und, die in diesem Dockerfile-Beispiel verwendet werden, sind nach dem Dockerfile aufgeführt.

```
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.tar.gz /usr/local/velocity/installation/gapwalk/
gapwalk.tar.gz

# 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"]
```

Im Folgenden finden Sie den Inhalt von `install-gapwalk.sh`

```
#!/bin/sh

# Vars
TEMP_DIR=/bluage/tomcat.gapwalk/temp

# Install
echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk-bluage.tar.gz ${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/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk-bluage.tar.gz -C /bluage/tomcat.gapwalk
# Remove temporary extraction dir
sudo rm -rf ${TEMP_DIR}
```

Das Folgende ist der Inhalt von `install-app.sh`.

```
#!/bin/sh

APP_DIR=/workdir/apps
TOMCAT_GAPWALK_DIR=/bluage/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/
```

2. Geben Sie die Verbindungsinformationen für die Datenbank, die Sie als Teil der Voraussetzungen im folgenden Codeausschnitt erstellt haben, in die `application-main.yml` Datei ein, die sich im Ordner befindet. `{TOMCAT_GAPWALK_DIR}/config` Weitere Informationen finden Sie unter [Erstellen eines Aurora PostgreSQL-DB-Clusters und Herstellen einer Verbindung zu diesem](#).

```
datasource:
  jicsDs:
    driver-class-name :
    url:
    username:
    password:
    type :
```

3. Erstellen Sie das Image und übertragen Sie es in Ihr Amazon ECR-Repository. Anweisungen finden Sie unter [Pushing a Docker-Image](#) im Amazon Elastic Container Registry-Benutzerhandbuch. Erstellen Sie dann je nach Situation einen Amazon EKS-Pod oder eine Amazon ECS-Aufgabendefinition mit Ihrem Amazon ECR-Image und stellen Sie es in Ihrem Cluster bereit. Weitere Informationen finden Sie beispielsweise unter [Erstellen einer Aufgabendefinition mithilfe der Konsole](#) im Amazon Elastic Container Service Developer Guide und [Bereitstellen einer Beispielanwendung](#) im Amazon EKS-Benutzerhandbuch.

## Testen Sie die bereitgestellte Anwendung

Ein Beispiel für das Testen der PlanetsDemo Anwendung finden Sie unter [the section called “Testen Sie die Anwendung PlanetsDemo”](#).

## Aktualisierung der AWS Blu Age Runtime auf Amazon EC2

In diesem Handbuch wird beschrieben, wie Sie die AWS Blu Age Runtime auf Amazon EC2 aktualisieren.

### Themen

- [Voraussetzungen](#)
- [Aktualisieren Sie die AWS Blu Age Runtime in der Amazon EC2 EC2-Instance](#)
- [Aktualisieren Sie die AWS Blu Age Runtime in einem Container](#)

### Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- Informationen darüber, ob es spezielle Anweisungen für Ihre Version gibt, finden Sie unter [the section called “AWS Anweisungen zum Upgrade auf Blu Age”](#).
- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) Vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Stellen Sie sicher, dass Sie über eine Amazon EC2 EC2-Instance verfügen, die die neueste AWS Blu Age Runtime enthält. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon-EC2-Linux-Instanzen](#).
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur Amazon EC2 EC2-Instance herstellen können, z. B. mithilfe von SSM.
- Laden Sie die Version der AWS Blu Age Runtime herunter, auf die Sie ein Upgrade durchführen möchten. Weitere Informationen finden Sie unter [the section called “AWS Einrichtung von Blu Age Runtime \(nicht verwaltet\)”](#) Das Framework besteht aus zwei Binärdateien: `aws-bluage-runtime-x.x.x.x.tar.gz` und `aws-bluage-webapps-x.x.x.x.tar.gz`.

### Aktualisieren Sie die AWS Blu Age Runtime in der Amazon EC2 EC2-Instance


Führen Sie die folgenden Schritte aus, um die AWS Blu Age Runtime zu aktualisieren.

1. Connect zu Ihrer Amazon EC2 EC2-Instance her und ändern Sie den Benutzer in su, indem Sie den folgenden Befehl ausführen.

```
sudo su
```

Sie benötigen Superuser-Rechte, um die Befehle in diesem Tutorial ausführen zu können.

2. Erstellen Sie zwei Ordner, einen für jede Binärdatei.
3. Benennen Sie jeden Ordner mit demselben Namen wie die Binärdatei.
4. Kopieren Sie jede Binärdatei in den entsprechenden Ordner.

 Warning

Beim Extrahieren jeder Binärdatei wird ein Ordner mit demselben Namen erzeugt. Wenn Sie also beide Binärdateien nacheinander am selben Speicherort extrahieren, überschreiben Sie den Inhalt.

5. Verwenden Sie die folgenden Befehle, um die Binärdateien zu extrahieren. Führen Sie die Befehle in jedem Ordner aus.

```
tar xvf aws-bluage-runtime-x.x.x.x.tar.gz
tar xvf aws-bluage-webapps-x.x.x.x.tar.gz
```

6. Stoppen Sie die Apache Tomcat-Dienste mit den folgenden Befehlen.

```
systemctl stop tomcat.service
systemctl stop tomcat-webapps.service
```

7. Ersetzen Sie den Inhalt von <your-tomcat-path>/shared/ durch den Inhalt von. aws-bluage-runtime-x.x.x.x/velocity/shared/
8. Ersetzen Sie <your-tomcat-path>/webapps/gapwalk-application.war durch aws-bluage-runtime-x.x.x.x/velocity/webapps/gapwalk-application.war.
9. Ersetzen Sie die Kriegsdateien in<your-tomcat-path>/webapps/, nämlich bac.war undjac.war, durch dieselben Dateien vonaws-bluage-webapps-x.x.x.x/velocity/webapps/.
10. Starten Sie die Apache Tomcat-Dienste, indem Sie die folgenden Befehle ausführen.

```
systemctl start tomcat.service
```

```
systemctl start tomcat-webapps.service
```

## 11. Überprüfen Sie die Protokolle.

Führen Sie die folgenden Befehle aus, um den Status der bereitgestellten Anwendung zu überprüfen.

```
curl http://localhost:8080/gapwalk-application/
```

Die folgende Meldung sollte angezeigt werden.

```
Jics application is running
```

```
curl http://localhost:8181/jac/api/services/rest/jicsservice/
```

Die folgende Meldung sollte angezeigt werden.

```
Jics application is running
```

```
curl http://localhost:8181/bac/api/services/rest/bluesamserver/serverIsUp
```

Die Antwort sollte leer sein.

Die AWS Blu Age-Laufzeit wurde erfolgreich aktualisiert.

Aktualisieren Sie die AWS Blu Age Runtime in einem Container

Führen Sie die folgenden Schritte aus, um die AWS Blu Age Runtime zu aktualisieren.

1. Erstellen Sie Ihr Docker-Image mit der gewünschten AWS Blu Age Runtime-Version neu. Anweisungen finden Sie unter [the section called “Einrichtung von AWS Blu Age Runtime \(nicht verwaltet\) auf Amazon EC2”](#).
2. Übertragen Sie Ihr Docker-Image in Ihr Amazon ECR-Repository.
3. Beenden Sie Ihren Amazon ECS- oder Amazon EKS-Service und starten Sie ihn neu.
4. Überprüfen Sie die Protokolle.

Die AWS Blu Age Runtime wurde erfolgreich aktualisiert.

## AWS Blu Age Runtime (auf Amazon EC2) Amazon Alarms CloudWatch

Damit Sie besser informiert werden, wenn Ihre bereitgestellten Anwendungen auf Ausnahmen stoßen, die für Ihre Anwendung eine Übergangsfrist bedeuten, können Sie den Empfang Ihres Anwendungsprotokolls einrichten CloudWatch und einen Alarm hinzufügen, der Sie vor möglichen Fehlern warnt.

### Bereitstellung der CloudWatch Protokollierung

Standardmäßig enthält die `application-main.yml` Datei einen Verweis auf eine andere Logging-Konfigurationsdatei mit dem Namen `logback-cloudwatch.yml`.

```
logging:
  config: classpath:logback-cloudwatch.xml
```

Beide Dateien befinden sich im Konfigurationsordner. So wird die CloudWatch Protokollierung konfiguriert, wie in den folgenden Abschnitten erklärt.

### Konfiguration der CloudWatch Protokollierung

Die `logback-cloudwatch.xml` Standarddatei hat den folgenden Inhalt.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration>
<configuration>

  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </encoder>
  </appender>

  <appender name="cloudwatch"
class="com.netfactive.bluage.runtime.cloudwatchlogger.CloudWatchAppender">
    <logGroup>BluAgeRuntimeOnEC2-Logs</logGroup>
    <logStream>%date{yyyy-MM-dd,UTC}.%instanceId.%uuid</logStream>
    <layout>
      <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC} %level --- [%thread{15}]
%logger{40} : %msg%n%xThrowable</pattern>
    </layout>
    <appender-ref ref="console" />
```

```
</appender>

<root level="INFO">
  <appender-ref ref="cloudwatch" />
</root>
</configuration>
```

Alles außerhalb des `<appender name="cloudwatch"/>` Elements ist die Standard-Logback-Konfiguration. Diese Datei enthält zwei Appender: einen Konsolen-Appender zum Senden von Protokollen an die Konsole und einen CloudWatch Appender zum Senden von Protokollen. CloudWatch

Das `level` Attribut im `root` Element gibt die Protokollierungsebene der gesamten Anwendung an.

Die erforderlichen Werte innerhalb des Tags `<appender name="cloudwatch"/>` sind:

- `<logGroup/>`: Legt den Namen der Protokollgruppe fest. CloudWatch Wenn der Wert nicht angegeben ist, ist er standardmäßig. `BluAgeRuntimeOnEC2-Logs` Wenn die Protokollgruppe nicht existiert, wird sie automatisch erstellt. Dieses Verhalten kann durch die Konfiguration geändert werden, auf die weiter unten eingegangen wird.
- `<logStream/>`: Legt den Namen des LogStream (innerhalb der Protokollgruppe) fest. CloudWatch

Optionale Werte:

- `<region/>`: Überschreibt die Region, in die der Log-Stream geschrieben wird. Standardmäßig werden Protokolle in dieselbe Region wie die EC2-Instance gesendet.
- `<layout/>`: Das Muster, das die Protokollnachrichten verwenden werden.
- `<maxbatchsize/>`: Die maximale Anzahl von Protokollnachrichten, an die CloudWatch pro Vorgang gesendet werden können.
- `<maxbatchtimemillis/>`: Die Zeit in Millisekunden, die für das Schreiben von CloudWatch Protokollen benötigt wird.
- `<maxqueuewaittimemillis/>`: Die Zeit in Millisekunden, in der versucht wird, Anfragen in die interne Protokollwarteschlange einzufügen.
- `<internalqueuesize/>`: Die maximale Größe der internen Warteschlange.
- `<createlogdests/>`: Erstellt eine Protokollgruppe und einen Protokollstream, falls sie nicht existieren.
- `<initialwaittimemillis/>`: Die Zeitspanne, für die der Thread beim Start in den Ruhezustand versetzt werden soll. Diese anfängliche Wartezeit ermöglicht das anfängliche Sammeln von Protokollen.



- `<maxeventmessagesize/>`: Die maximale Größe eines Protokollereignisses. Protokolle, die diese Größe überschreiten, werden nicht gesendet.
- `<truncateeventmessages/>`: Kürzt zu lange Nachrichten.
- `<printrejectedevents/>`: Aktivieren Sie den Notfall-Appender.

## CloudWatch einrichten

Damit die obige Konfiguration Logs korrekt überträgt CloudWatch, aktualisieren Sie Ihre Amazon EC2 IAM-Instance-Profilrolle, um ihr zusätzliche Berechtigungen für die BluAgeRuntimeOn `EC2-Logs`-Protokollgruppe und ihre Logstreams zu gewähren:

- `logs:CreateLogStream`
- `logs:DescribeLogStreams`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`
- `logs:DescribeLogGroups`

## Einrichtung des Alarms

Dank der CloudWatch Protokolle können Sie dann je nach Ihrer Anwendung und Ihren Anforderungen verschiedene Metriken und Alarmer konfigurieren. Insbesondere können Sie proaktive Alarmer für Nutzungswarnungen einrichten, um bei Fehlern gewarnt zu werden, die Ihre Anwendung in eine Übergangszeit bringen könnten (und letztendlich verhindern, dass sie überhaupt funktioniert). Um dies zu erreichen, können Sie den Protokollen eine Metrik hinzufügen, die sich auf die Zeichenfolge „Fehler C5001“ bezieht. Dadurch werden Fehler bei der Verbindung zum AWS Blu Age-Kontrollsystem hervorgehoben. Anschließend können Sie einen Alarm definieren, der auf diese Metrik reagiert.

## Einrichten lizenzierter Abhängigkeiten in AWS Blu Age Runtime auf Amazon EC2

In diesem Handbuch wird beschrieben, wie Sie zusätzliche lizenzierte Abhängigkeiten einrichten, die Sie mit AWS Blu Age Runtime auf Amazon EC2 verwenden können.

### Themen

- [Voraussetzungen](#)
- [Übersicht](#)

- [Richten Sie die Abhängigkeiten für JAC- und BAC-Webapps ein](#)

## Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- [the section called “AWS Voraussetzungen für die Laufzeit von Blu Age”](#) Vollständig und [the section called “AWS Einführung in Blu Age Runtime”](#).
- Stellen Sie sicher, dass Sie über eine Amazon EC2-Instance verfügen, die die neueste AWS Blu Age Runtime (auf Amazon EC2) enthält. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon-EC2-Linux-Instanzen](#).
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur Amazon EC2 EC2-Instance herstellen können, z. B. mithilfe von SSM.
- Holen Sie sich die folgenden Abhängigkeiten aus ihren Quellen.

## Oracle-Datenbank

Stellen Sie einen [Oracle-Datenbanktreiber bereit](#). Wir haben die Funktionalität der AWS Blu Age Runtime (auf Amazon EC2) mit der Version ojdbc11-23.3.0.23.09.jar getestet, aber eine neuere Version könnte kompatibel sein.

## IBM MQ-Verbindung

Stellen Sie einen [IBM MQ-Client bereit](#). Wir haben die Funktionalität der AWS Blu Age Runtime (auf Amazon EC2) mit der Version com.ibm.mq.jakarta.client-9.3.4.1.jar getestet, aber eine neuere Version könnte kompatibel sein.

Stellen Sie mit dieser Abhängigkeitsversion auch die folgenden transitiven Abhängigkeiten bereit:

- bcprov-jdk15to18-1.76.jar
- bcpkix-jdk15to18-1.76.jar
- bcutil-jdk15to18-1.76.jar

## DDS-Druckerdateien

Stellen Sie die [Jasper-Berichtsbibliothek bereit](#). Wir haben die Funktionalität von AWS Blu Age Runtime (auf Amazon EC2) mit jasperreports-6.16.0.jar getestet, aber eine neuere Version könnte kompatibel sein.

Stellen Sie mit dieser Abhängigkeitsversion auch die folgenden transitiven Abhängigkeiten bereit:

- 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-sammlungen-3.2.2.jar

## Übersicht

Gehen Sie wie folgt vor, um die Abhängigkeiten zu installieren.

1. Connect zu Ihrer Amazon EC2 EC2-Instance her und ändern Sie den Benutzer in su, indem Sie den folgenden Befehl ausführen.

```
sudo su
```

Sie benötigen Superuser-Rechte, um die Befehle in diesem Tutorial ausführen zu können.

2. Navigieren Sie zum Verzeichnis `<your-tomcat-path>/extra/`.

```
cd <your-tomcat-path>/extra/
```

3. Kopieren Sie alle oben genannten Abhängigkeiten nach Bedarf in diesen Ordner.
4. Stoppen und starten Sie tomcat.service, indem Sie die folgenden Befehle ausführen.

```
systemctl stop tomcat.service
```

```
systemctl start tomcat.service
```

5. Überprüfen Sie den Status des Dienstes, um sicherzustellen, dass er ausgeführt wird.

```
systemctl status tomcat.service
```

## 6. Überprüfen Sie die Protokolle.

Richten Sie die Abhängigkeiten für JAC- und BAC-Webapps ein

1. Wenn Ihre JICS- oder Blusam-Datenbank auf Oracle gehostet wird, müssen Sie den Oracle-Datenbanktreiber in bereitstellen. `<your-tomcat-path>/extra`
2. Erstellen Sie den Ordner, falls er noch nicht vorhanden ist.
3. Stoppen Sie Ihren Apache Tomcat-Server und starten Sie ihn neu.
4. Überprüfen Sie die Protokolle.

## Testen Sie die Anwendung PlanetsDemo

Um den Status der bereitgestellten PlanetsDemo Anwendung zu überprüfen, führen Sie die folgenden Befehle aus, nachdem Sie `load-balancer-dns-name`, ersetzt haben `listener-port`, und `web-binary-name` zwar mit den richtigen Werten für Ihr Setup.

```
curl http://load-balancer-dns-name:listener-port/gapwalk-application/
```

Wenn die Anwendung ausgeführt wird, wird die folgende Ausgabemeldung angezeigt: `Jics application is running`.

Führen Sie als Nächstes den folgenden Befehl aus.

```
curl http://load-balancer-dns-name:listener-port/jac/api/services/rest/jicsservice/
```

Wenn die Anwendung ausgeführt wird, wird die folgende Ausgabenachricht angezeigt: `Jics application is running`.

```
Jics application is running
```

Wenn Sie Blusam konfiguriert haben, können Sie mit einer leeren Antwort rechnen, wenn Sie den folgenden Befehl ausführen.

```
curl http://load-balancer-DNS-name:listener-port/bac/api/services/rest/bluesamserver/  
serverIsUp
```

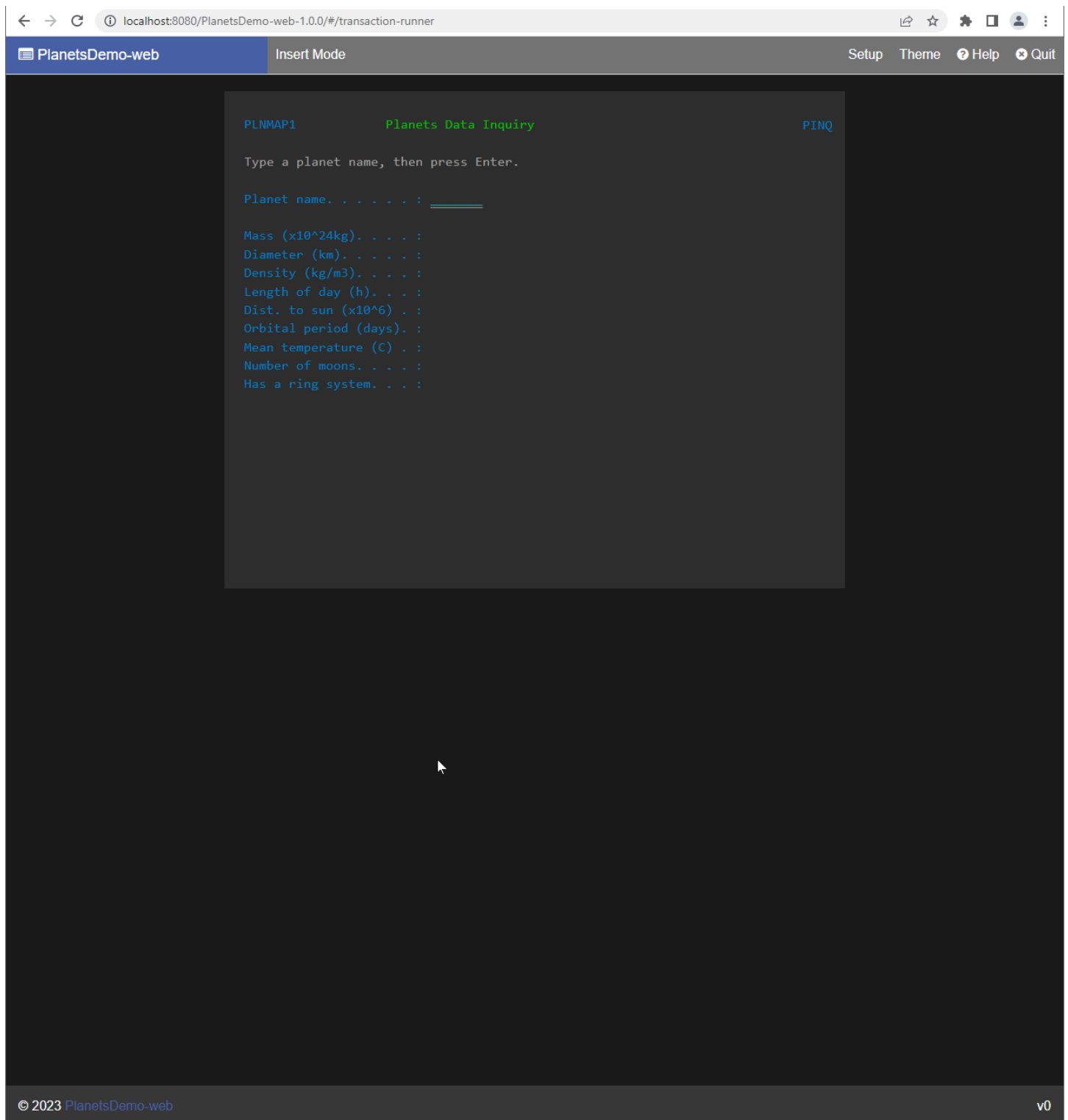
Notieren Sie sich den Namen der Web-Binärdatei (PlanetsDemo-web-1.0.0, falls unverändert).  
Verwenden Sie eine URL des PlanetsDemo folgenden Formats, um auf die Anwendung zuzugreifen.

```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

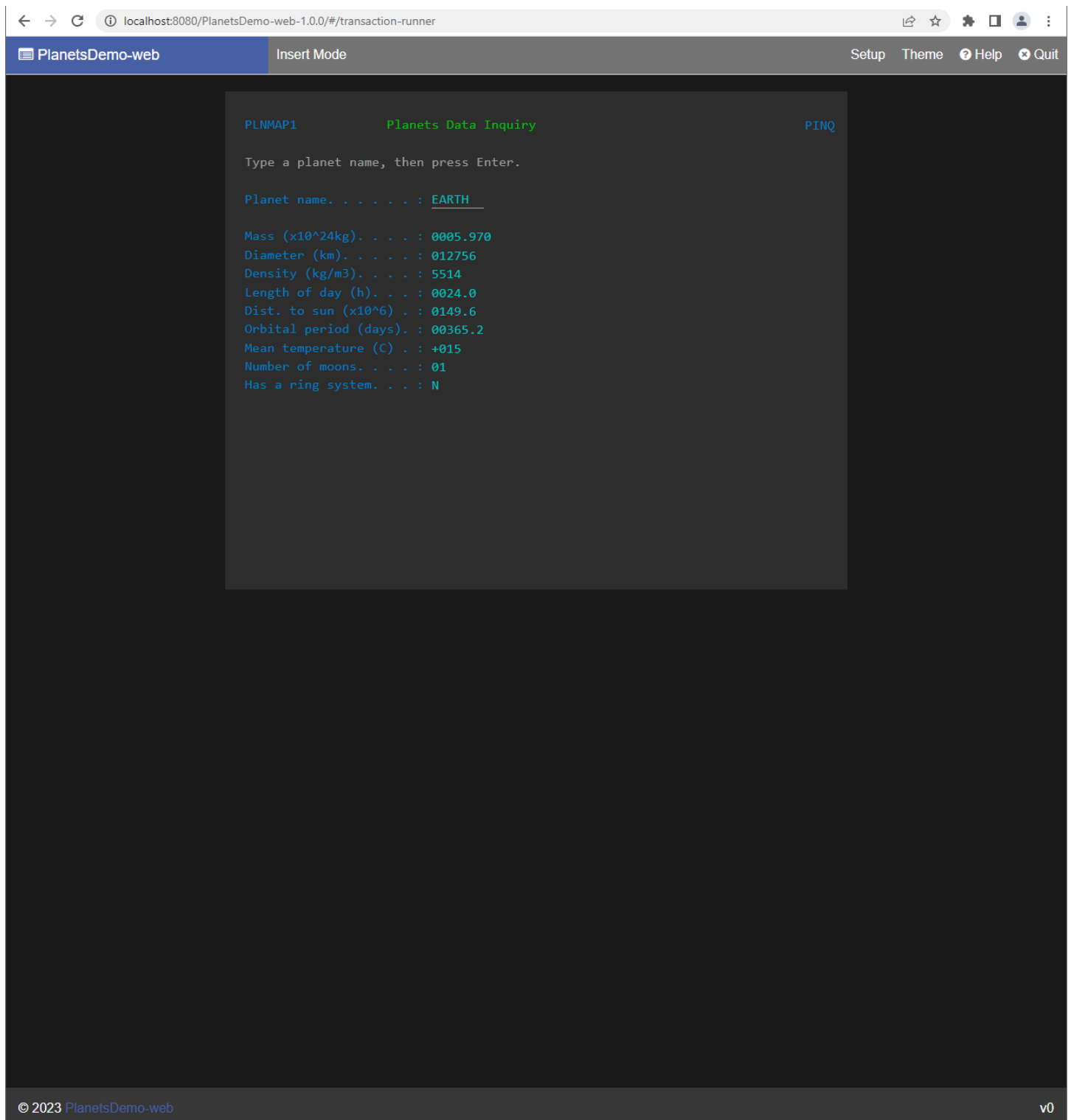
Nach dem Start der PlanetsDemo Anwendung wird die Startseite angezeigt.



Geben Sie PINQ in das Textfeld ein und drücken Sie dann die Eingabetaste. Die Datenabfrageseite wird angezeigt.



Geben Sie beispielsweise EARTH in das PlanetsDemo Namensfeld ein, und drücken Sie dann die EINGABETASTE. Die Seite für den Planeten, den Sie eingegeben haben, wird angezeigt.



The screenshot shows a web browser window with the address bar displaying `localhost:8080/PlanetsDemo-web-1.0.0/#/transaction-runner`. The browser's title bar includes "PlanetsDemo-web" and "Insert Mode". The main content area displays a terminal window with the following text:

```
PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

Planet name. . . . . : EARTH

Mass (x10^24kg). . . . . : 0005.970
Diameter (km). . . . . : 012756
Density (kg/m3). . . . . : 5514
Length of day (h). . . . . : 0024.0
Dist. to sun (x10^6). . . . . : 0149.6
Orbital period (days). . . . . : 00365.2
Mean temperature (C) . . . . . : +015
Number of moons. . . . . : 01
Has a ring system. . . . . : N
```

At the bottom left of the browser window, there is a copyright notice: "© 2023 PlanetsDemo-web". At the bottom right, the version number "v0" is displayed.

## Ändern Sie den Quellcode mit Blu Age Developer IDE

Wenn Sie die von AWS-managed AWS Blu Age Runtime Engine verwenden, können Sie Blu Age Developer verwenden, um den generierten Quellcode zu ändern. Möglicherweise möchten Sie dies

tun, wenn Sie den modernisierten Code aus irgendeinem Grund aktualisieren müssen oder wenn ein Teil des alten Quellcodes nicht modernisiert werden konnte. Sie greifen über Amazon AppStream 2.0 auf Blu Age Developer zu. In diesem Abschnitt wird beschrieben, wie Sie Blu Age Developer auf AppStream 2.0 einrichten. Außerdem wird anhand der Beispielanwendung erklärt, wie Blu Age Developer zum Aktualisieren des Quellcodes verwendet wird PlanetsDemo.

## Themen

- [Tutorial: AppStream 2.0 für AWS Blu Age Developer IDE einrichten](#)
- [Tutorial: Verwenden Sie AWS Blu Age Developer auf AppStream 2.0](#)

## Tutorial: AppStream 2.0 für AWS Blu Age Developer IDE einrichten

AWS Mainframe Modernization bietet über Amazon AppStream 2.0 mehrere Tools. AppStream 2.0 ist ein vollständig verwalteter, sicherer Anwendungs-Streaming-Dienst, mit dem Sie Desktop-Anwendungen an Benutzer streamen können, ohne Anwendungen neu schreiben zu müssen. AppStream 2.0 bietet Benutzern sofortigen Zugriff auf die Anwendungen, die sie benötigen, und bietet eine reaktionsschnelle, flüssige Benutzererfahrung auf dem Gerät ihrer Wahl. Die Verwendung von AppStream 2.0 zum Hosten von Runtime Engine-spezifischen Tools gibt den Anwendungsteams von Kunden die Möglichkeit, die Tools direkt von ihren Webbrowsern aus zu verwenden und mit Anwendungsdateien zu interagieren, die entweder in Amazon S3 S3-Buckets oder Repositories gespeichert sind. CodeCommit

Informationen zur Browserunterstützung in AppStream 2.0 finden Sie unter [Systemanforderungen und Funktionsunterstützung \(Webbrowser\)](#) im Amazon AppStream 2.0-Administrationshandbuch. Wenn Sie bei der Verwendung von AppStream 2.0 Probleme haben, finden Sie weitere Informationen unter [Troubleshooting AppStream 2.0-Benutzerprobleme](#) im Amazon AppStream 2.0-Administrationshandbuch.

In diesem Dokument wird beschrieben, wie Sie die AWS Blu Age Developer IDE auf einer AppStream 2.0-Flotte einrichten.

## Themen

- [Voraussetzung](#)
- [Schritt 1: Einen Amazon-S3-Bucket erstellen](#)
- [Schritt 2: Hängen Sie eine Richtlinie an den S3-Bucket an](#)
- [Schritt 3: Dateien in den Amazon S3 S3-Bucket hochladen](#)



- [Schritt 4: Vorlagen herunterladen AWS CloudFormation](#)
- [Schritt 5: Erstellen Sie die Flotte mit AWS CloudFormation](#)
- [Schritt 6: Greifen Sie auf eine Instanz zu](#)
- [Bereinigen von -Ressourcen](#)

## Voraussetzung

Laden Sie die [Archivdatei](#) herunter, die die Artefakte enthält, die Sie für die Einrichtung der AWS Blu Age Developer IDE unter AppStream 2.0 benötigen.

### Note

Das ist eine große Datei. Wenn Sie Probleme mit dem Timeout des Vorgangs haben, empfehlen wir die Verwendung einer Amazon EC2 EC2-Instance, um die Upload- und Download-Leistung zu verbessern.

## Schritt 1: Einen Amazon-S3-Bucket erstellen

Erstellen Sie einen Amazon S3 S3-Bucket in derselben Version AWS-Region wie die AppStream 2.0-Flotte, die Sie erstellen werden. Dieser Bucket enthält die Artefakte, die Sie benötigen, um dieses Tutorial abzuschließen.

## Schritt 2: Hängen Sie eine Richtlinie an den S3-Bucket an

Hängen Sie die folgende Richtlinie an den Bucket an, den Sie für dieses Tutorial erstellen. Stellen Sie sicher, dass Sie es MYBUCKET durch den tatsächlichen Namen des Buckets ersetzen, den Sie erstellen.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "AllowAppStream2.0ToRetrieveObjects",
    "Effect": "Allow",
    "Principal": {
      "Service": "appstream.amazonaws.com"
    },
    "Action": "s3:GetObject",
```

```
    "Resource": "arn:aws:s3:::MYBUCKET/*"  
  }]  
}
```

### Schritt 3: Dateien in den Amazon S3 S3-Bucket hochladen

Entpacken Sie die Dateien, die Sie unter Voraussetzung heruntergeladen haben, und laden Sie den `appstream` Ordner in Ihren Bucket hoch. Durch das Hochladen dieses Ordners wird die richtige Struktur in Ihrem Bucket erstellt. Weitere Informationen finden Sie unter [Hochladen von Objekten](#) im Amazon S3 S3-Benutzerhandbuch.

### Schritt 4: Vorlagen herunterladen AWS CloudFormation

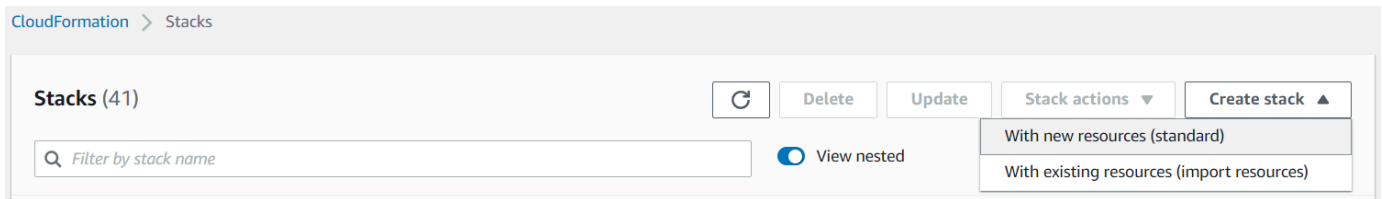
Laden Sie die folgenden AWS CloudFormation Vorlagen herunter. Sie benötigen diese Vorlagen, um die AppStream 2.0-Flotte zu erstellen und auszufüllen.

- [cfn-m2-.yaml appstream-elastic-fleet-linux](#)
- [appstream-bluage-dev-toolscfn-m2--linux.yaml](#)
- [appstream-bluage-shared-linuxcfn-m2-.yaml](#)
- [appstream-chrome-linuxcfn-m2-.yaml](#)
- [appstream-eclipse-jee-linuxcfn-m2-.yaml](#)
- [appstream-pgadmin-linuxcfn-m2-.yaml](#)

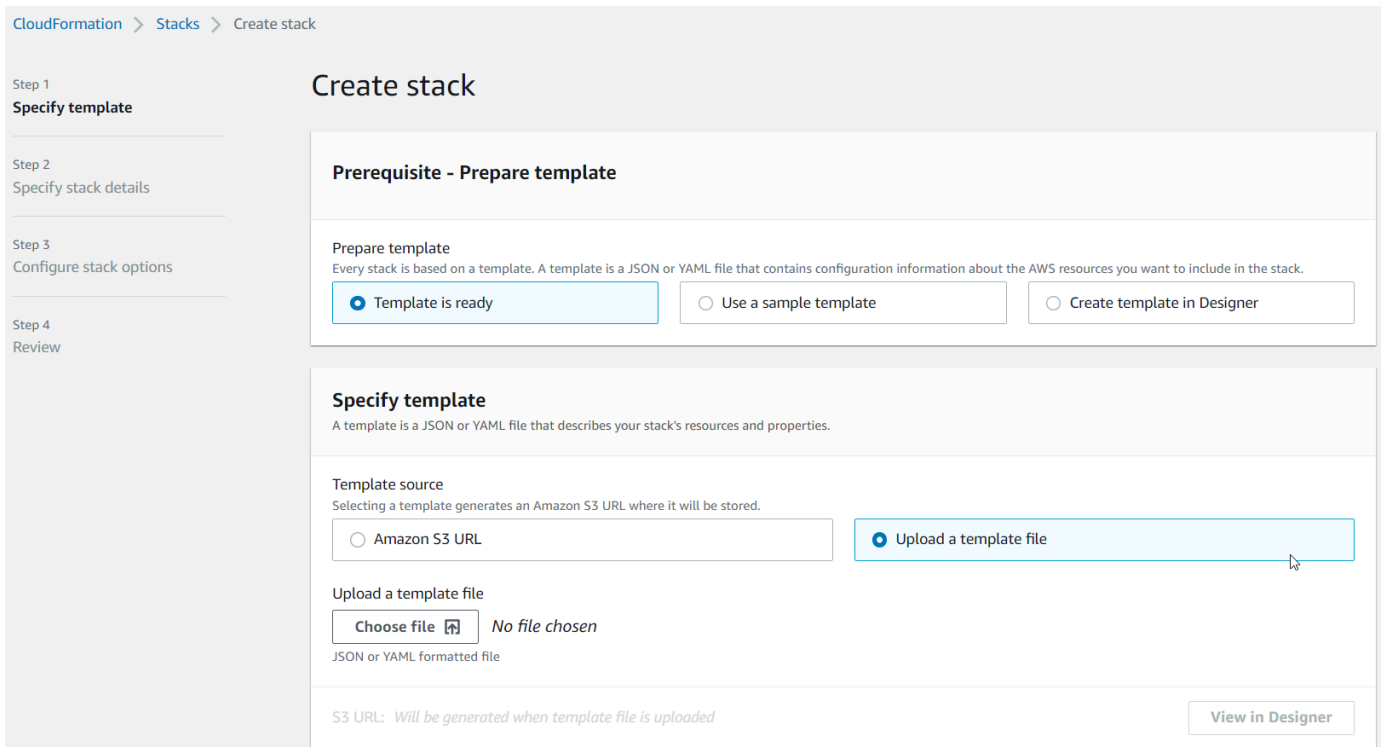
### Schritt 5: Erstellen Sie die Flotte mit AWS CloudFormation

In diesem Schritt verwenden Sie die `cfn-m2-appstream-elastic-fleet-linux.yaml` AWS CloudFormation Vorlage, um eine AppStream 2.0-Flotte und einen Stack zum Hosten der AWS Blu Age Developer IDE zu erstellen. Nachdem Sie die Flotte und den Stack erstellt haben, führen Sie die anderen AWS CloudFormation Vorlagen aus, die Sie im vorherigen Schritt heruntergeladen haben, um die Entwickler-IDE und andere erforderliche Tools zu installieren.

1. Navigieren Sie AWS CloudFormation in der AWS Managementkonsole zu und wählen Sie Stacks aus.
2. Wählen Sie unter Stacks die Optionen Stapel erstellen und Mit neuen Ressourcen (Standard) aus:



- Wählen Sie unter Stapel erstellen die Option Vorlage ist bereit und Laden Sie eine Vorlagendatei hoch:



- Wählen Sie „Datei auswählen“ und navigieren Sie zur Datei `icfn-m2-appstream-elastic-fleet-linux.yaml`. Wählen Sie Weiter aus.
- Geben Sie im Feld Stapeldetails an die folgenden Informationen ein:
  - Ein Name für den Stack.
  - Ihre Standardsicherheitsgruppe und zwei Subnetze dieser Sicherheitsgruppe.

#### Note

Die beiden Subnetze der Sicherheitsgruppe müssen sich in unterschiedlichen Verfügbarkeitszonen befinden.

- Wählen Sie Weiter und dann erneut Weiter.

7. Wählen Sie Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden. , und wählen Sie dann Absenden.
8. Nachdem Sie die Flotte erstellt haben, erstellen Sie CloudFormation Stapel mit den anderen heruntergeladenen Vorlagen, um die Einrichtung der Anwendungen abzuschließen. Stellen Sie sicher, dass Sie BucketNamejedes Mal aktualisieren, um auf den richtigen S3-Bucket zu verweisen. Sie können das BucketNamein der CloudFormation Konsole bearbeiten. Alternativ können Sie die Vorlagendateien direkt bearbeiten und die S3Bucket Eigenschaft aktualisieren.

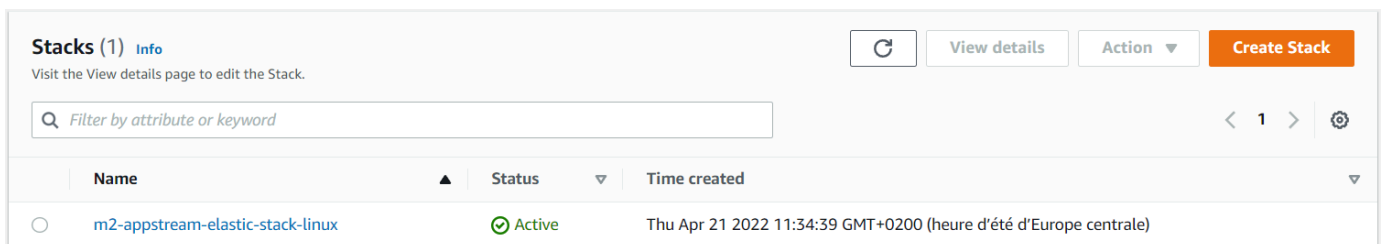
### Note

Die heruntergeladenen Vorlagen erwarten, dass sie Assets in einem S3-Bucket mit einer Ordnerstruktur namens `findenappstream/bluage/developer-ide/`. Der Bucket muss sich in derselben Flotte AWS-Region befinden, die Sie erstellt haben.

## Schritt 6: Greifen Sie auf eine Instanz zu

Nachdem Sie die Flotte erstellt und gestartet haben, können Sie einen temporären Link erstellen, um über den nativen Client auf die Flotte zuzugreifen.

1. Navigieren Sie im zu AppStream 2.0 AWS Management Console und wählen Sie den zuvor erstellten Stapel aus:



The screenshot shows the AWS AppStream 2.0 Stacks console. At the top, it says "Stacks (1) Info" and "Visit the View details page to edit the Stack." There are buttons for "View details", "Action", and "Create Stack". Below that is a search bar with the placeholder "Filter by attribute or keyword". A table lists the stacks with columns for Name, Status, and Time created. One stack is listed: "m2-appstream-elastic-stack-linux" with a status of "Active" and a creation time of "Thu Apr 21 2022 11:34:39 GMT+0200 (heure d'été d'Europe centrale)".

Name	Status	Time created
m2-appstream-elastic-stack-linux	Active	Thu Apr 21 2022 11:34:39 GMT+0200 (heure d'été d'Europe centrale)

2. Wählen Sie auf der Seite mit den Stack-Details Aktion und anschließend Streaming-URL erstellen aus:

### Create Streaming URL: m2-appstream-elastic-stack-linux ✕

User ID \*

This is the User ID the URL will be associated to.

URL Expiration \*

Set the amount of time the URL will be active before expiration.

30 Minutes ▼

Cancel Get URL

3. Geben Sie unter Streaming-URL erstellen eine beliebige Benutzer-ID und eine URL-Ablaufzeit ein und wählen Sie dann URL abrufen aus. Sie erhalten eine URL, mit der Sie in einen Browser oder in den systemeigenen Client streamen können. Wir empfehlen, dass Sie in den nativen Client streamen.

## Bereinigen von -Ressourcen

Informationen zum Bereinigen des erstellten Stacks und der erstellten Flotten finden Sie unter [Eine AppStream 2.0-Flotte und einen Stack erstellen](#).

Wenn Sie die AppStream 2.0-Objekte gelöscht haben, können Sie oder der Kontoadministrator auch die S3-Buckets für Anwendungseinstellungen und Basisordner bereinigen.

### Note

Der Basisordner für einen bestimmten Benutzer ist für alle Flotten einzigartig. Daher müssen Sie ihn möglicherweise beibehalten, wenn andere AppStream 2.0-Stacks im selben Konto aktiv sind.

Sie können die AppStream 2.0-Konsole nicht zum Löschen von Benutzern verwenden. Stattdessen müssen Sie die Service-API mit dem verwenden AWS CLI. Weitere Informationen finden Sie unter [Benutzerpool-Administration](#) im Amazon AppStream 2.0-Administrationshandbuch.

## Tutorial: Verwenden Sie AWS Blu Age Developer auf AppStream 2.0

Dieses Tutorial zeigt Ihnen, wie Sie auf AWS Blu Age Developer auf AppStream 2.0 zugreifen und es mit einer Beispielanwendung verwenden, damit Sie die Funktionen ausprobieren können. Wenn Sie dieses Tutorial abgeschlossen haben, können Sie dieselben Schritte mit Ihren eigenen Anwendungen verwenden.

### Themen

- [Schritt 1: Erstellen einer Datenbank](#)
- [Schritt 2: Greifen Sie auf die Umgebung zu](#)
- [Schritt 3: Richten Sie die Laufzeit ein](#)
- [Schritt 4: Starten Sie die Eclipse-IDE](#)
- [Schritt 5: Richten Sie ein Maven-Projekt ein](#)
- [Schritt 6: Konfigurieren Sie einen Tomcat-Server](#)
- [Schritt 7: Auf Tomcat bereitstellen](#)
- [Schritt 8: Erstellen Sie die JICS-Datenbank](#)
- [Schritt 9: Starten und testen Sie die Anwendung](#)
- [Schritt 10: Debuggen Sie die Anwendung](#)
- [Bereinigen von -Ressourcen](#)

### Schritt 1: Erstellen einer Datenbank

In diesem Schritt verwenden Sie Amazon RDS, um eine verwaltete PostgreSQL-Datenbank zu erstellen, die die Demo-Anwendung zum Speichern von Konfigurationsinformationen verwendet.

1. Öffnen Sie die Amazon RDS-Konsole.
2. Wählen Sie Datenbanken > Datenbank erstellen.
3. Wählen Sie Standard create > PostgreSQL, behalten Sie die Standardversion bei und wählen Sie dann Kostenloses Kontingent.
4. Wählen Sie eine DB-Instance-ID aus.
5. Wählen Sie für Credential Settings die Option Masteranmeldedaten verwalten in AWS Secrets Manager. Weitere Informationen finden Sie unter [Passwortverwaltung mit Amazon RDS und AWS Secrets Manager](#) im Amazon-RDS-Benutzerhandbuch.

6. Stellen Sie sicher, dass es sich bei der VPC um dieselbe handelt, die Sie für die AppStream 2.0-Instance verwenden. Sie können Ihren Administrator nach diesem Wert fragen.
7. Wählen Sie für VPC-Sicherheitsgruppe die Option Create New aus.
8. Stellen Sie den öffentlichen Zugriff auf Ja ein.
9. Behalten Sie alle anderen Standardwerte bei. Überprüfen Sie diese Werte.
10. Wählen Sie Datenbank erstellen aus.

Um den Datenbankserver von Ihrer Instance aus zugänglich zu machen, wählen Sie den Datenbankserver in Amazon RDS aus. Wählen Sie unter Konnektivität und Sicherheit die VPC-Sicherheitsgruppe für den Datenbankserver aus. Diese Sicherheitsgruppe wurde zuvor für Sie erstellt und sollte eine ähnliche Beschreibung wie unter Erstellt mit der RDS-Managementkonsole haben. Wählen Sie Aktion > Regeln für eingehenden Datenverkehr bearbeiten, wählen Sie Regel hinzufügen und erstellen Sie eine Regel vom Typ PostgreSQL. Verwenden Sie als Regelquelle den Standard für die Sicherheitsgruppe. Sie können damit beginnen, den Quellnamen in das Feld Quelle einzugeben und dann die vorgeschlagene ID zu akzeptieren. Wählen Sie abschließend Regeln speichern aus.

## Schritt 2: Greifen Sie auf die Umgebung zu

In diesem Schritt greifen Sie auf die AWS Blu Age-Entwicklungsumgebung auf AppStream 2.0 zu.

1. Erkundigen Sie sich bei Ihrem Administrator nach der richtigen Methode für den Zugriff auf Ihre AppStream 2.0-Instanz. Allgemeine Informationen zu möglichen Clients und Konfigurationen finden Sie unter [AppStream 2.0 Access Methods and Clients](#) im Amazon AppStream 2.0-Administrationshandbuch. Ziehen Sie die Verwendung des nativen Clients in Betracht, um die beste Benutzererfahrung zu erzielen.
2. Wählen Sie in AppStream 2.0 Desktop.

## Schritt 3: Richten Sie die Laufzeit ein

In diesem Schritt richten Sie die AWS Blu Age-Laufzeit ein. Sie müssen die Runtime beim ersten Start und erneut einrichten, wenn Sie über ein Runtime-Upgrade informiert werden. Durch diesen Schritt wird Ihr .m2 Ordner aufgefüllt.

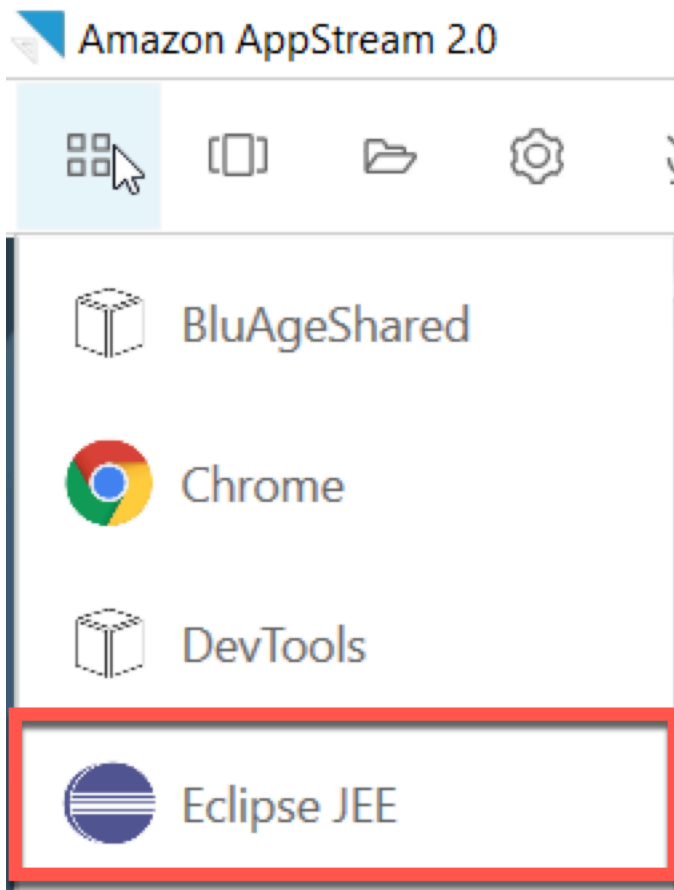
1. Wählen Sie in der Menüleiste Anwendungen und dann Terminal aus.
2. Geben Sie den folgenden Befehl ein:

```
~/_install-velocity-runtime.sh
```

## Schritt 4: Starten Sie die Eclipse-IDE

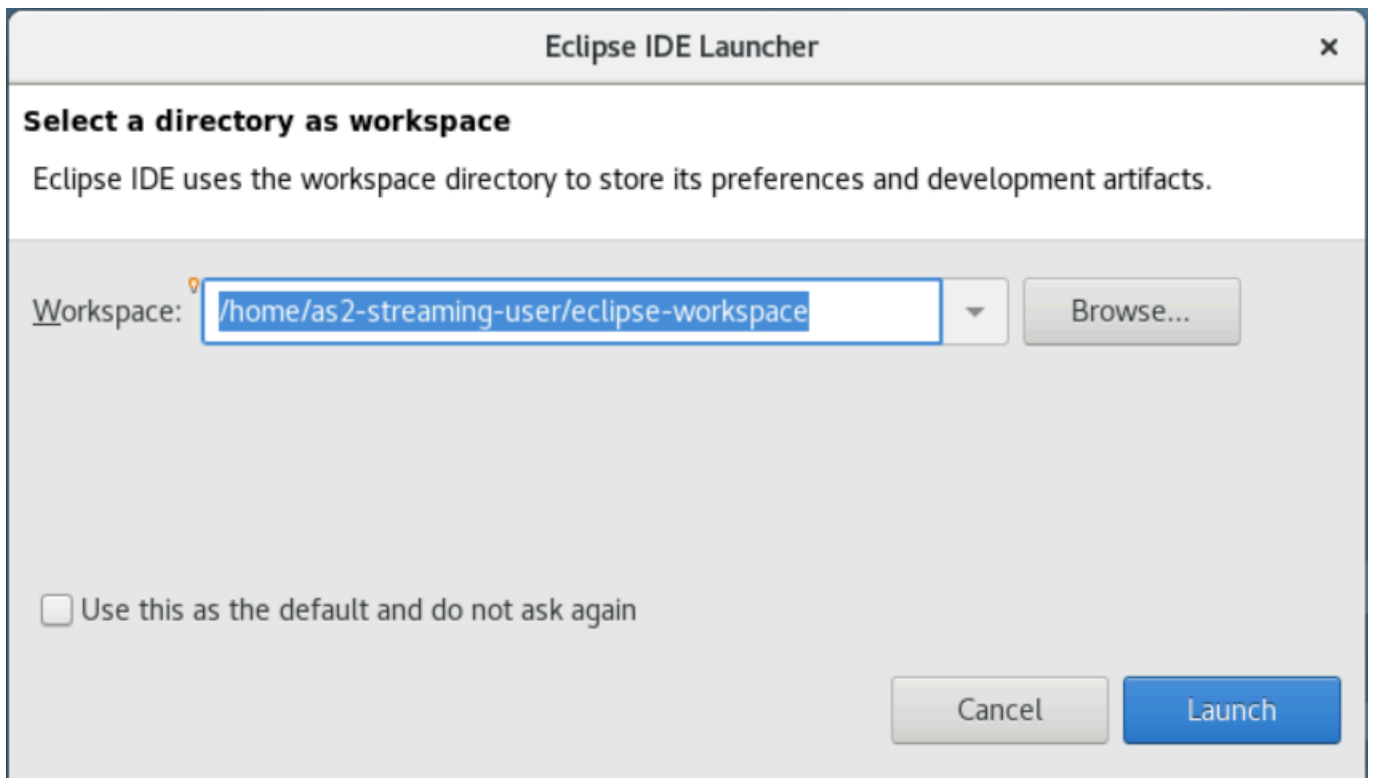
In diesem Schritt starten Sie die Eclipse-IDE und wählen einen Speicherort aus, an dem Sie einen Workspace erstellen möchten.

1. In AppStream 2.0 wählen Sie das Symbol „Anwendung starten“ in der Werkzeugleiste und wählen Sie dann Eclipse JEE.



2. Wenn der Launcher geöffnet wird, geben Sie den Ort ein, an dem Sie Ihren Workspace erstellen möchten, und wählen Sie Launch.





Optional können Sie Eclipse wie folgt von der Befehlszeile aus starten:

```
~/eclipse &
```

## Schritt 5: Richten Sie ein Maven-Projekt ein

In diesem Schritt importieren Sie ein Maven-Projekt für die Planets-Demo-Anwendung.

1. Laden Sie [PlanetsDemo-pom.zip](#) in Ihren Home-Ordner hoch. Sie können dazu die native Client-Funktion „Meine Dateien“ verwenden.
2. Verwenden Sie das `unzip` Befehlszeilentool, um die Dateien zu extrahieren.
3. Navigieren Sie in den entpackten Ordner und öffnen Sie das Stammverzeichnis `pom.xml` Ihres Projekts in einem Texteditor.
4. Bearbeiten Sie die `gapwalk.version` Eigenschaft so, dass sie der installierten AWS Blu Age-Laufzeit entspricht.

Wenn Sie sich nicht sicher sind, welche Version installiert ist, geben Sie den folgenden Befehl in einem Terminal ein:

```
cat ~/runtime-version.txt
```

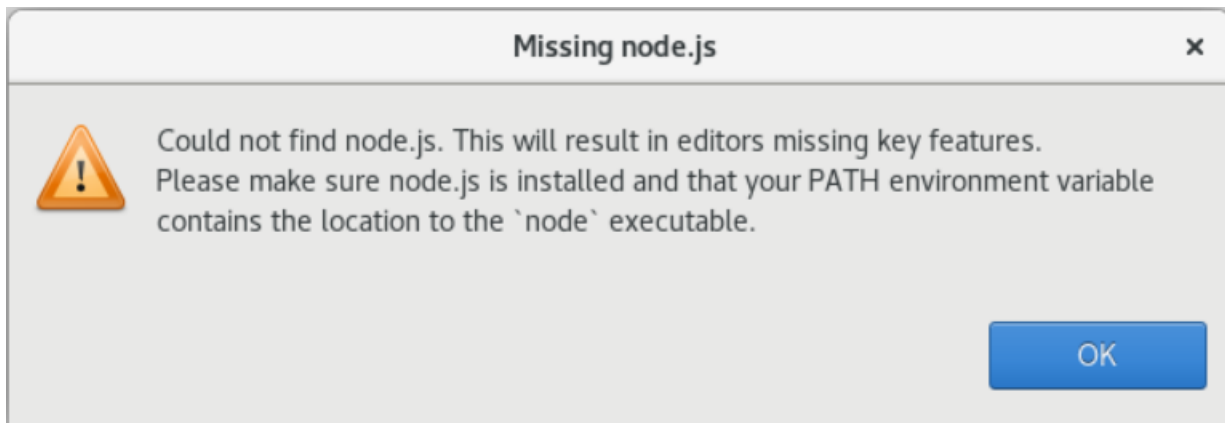
Dieser Befehl druckt die aktuell verfügbare Runtime-Version, zum Beispiel `3.1.0-b3257-dev`.

#### Note

Fügen Sie das `-dev` Suffix nicht in `gapwalk.version` ein. Ein gültiger Wert wäre `<gapwalk.version>3.1.0-b3257</gapwalk.version>` beispielsweise.

5. Wählen Sie in Eclipse „Datei“ und dann „Importieren“. Erweitern Sie im Dialogfenster Import die Option Maven und wählen Sie Existing Maven Projects aus. Wählen Sie Weiter aus.
6. Geben Sie unter Maven-Projekte importieren den Speicherort der extrahierten Dateien an und wählen Sie Fertig stellen.

Sie können das folgende Popup getrost ignorieren. Maven lädt eine lokale Kopie von `herunternode.js`, um den Angular-Teil (\*-Web) des Projekts zu erstellen:



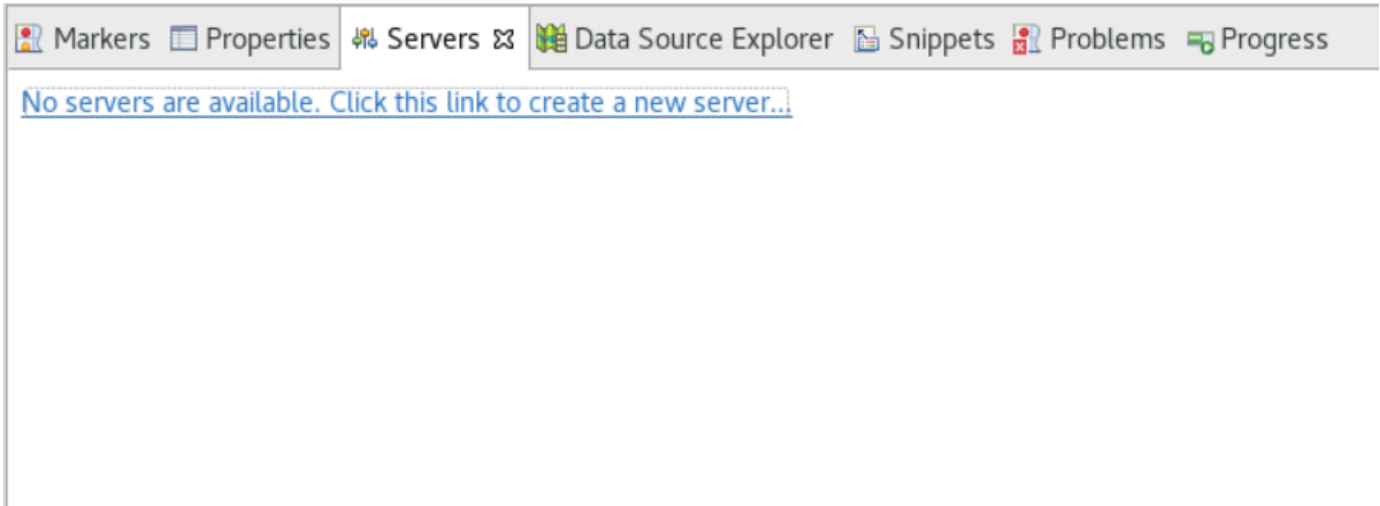
Warte bis zum Ende des Builds. Sie können den Build in der Fortschrittsansicht verfolgen.

7. Wählen Sie in Eclipse das Projekt aus und wählen Sie Ausführen als. Wählen Sie dann Maven install. Nachdem die Maven-Installation erfolgreich war, wird die `war` Datei unter `PlanetsDemoPom/PlanetsDemo-web/target/PlanetsDemo-web-1.0.0.war` erstellt.

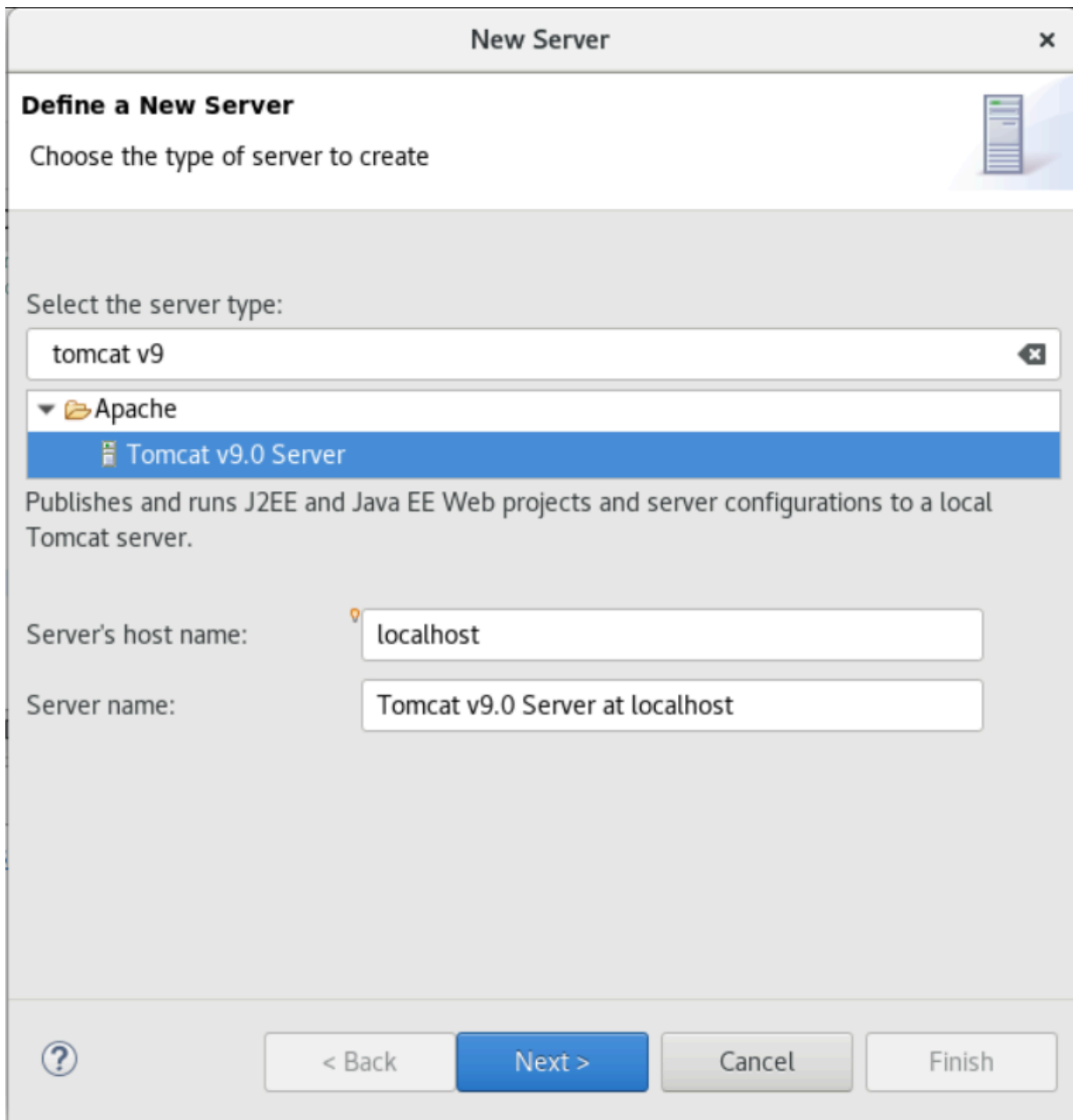
## Schritt 6: Konfigurieren Sie einen Tomcat-Server

In diesem Schritt konfigurieren Sie einen Tomcat-Server, auf dem Sie Ihre kompilierte Anwendung bereitstellen und starten.

1. Wählen Sie in Eclipse „Fenster“ > „Ansicht einblenden“ > „Server“, um die Serveransicht anzuzeigen:

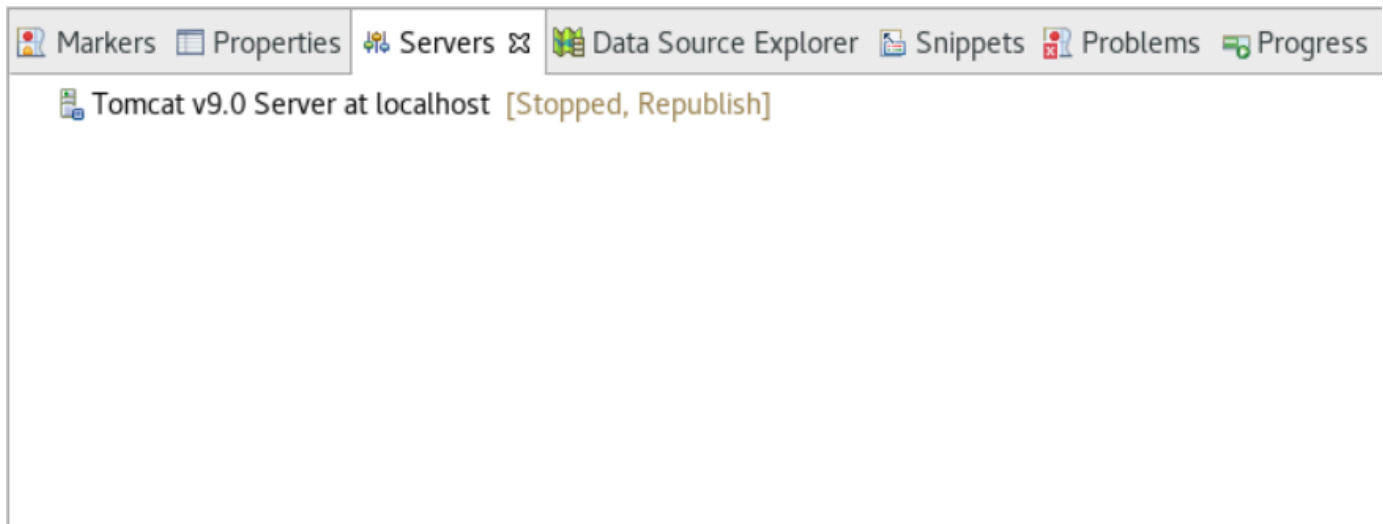


2. Wählen Sie Keine Server sind verfügbar. Klicken Sie auf diesen Link, um einen neuen Server zu erstellen... . Der Assistent für neue Server wird angezeigt. Geben Sie im Feld Servertyp auswählen des Assistenten tomcat v9 ein und wählen Sie Tomcat v9.0 Server aus. Wählen Sie anschließend Weiter.



3. Wählen Sie Durchsuchen und wählen Sie den Tomcat-Ordner im Stammverzeichnis des Home-Ordners aus. Behalten Sie für JRE den Standardwert bei und wählen Sie „Fertig stellen“.

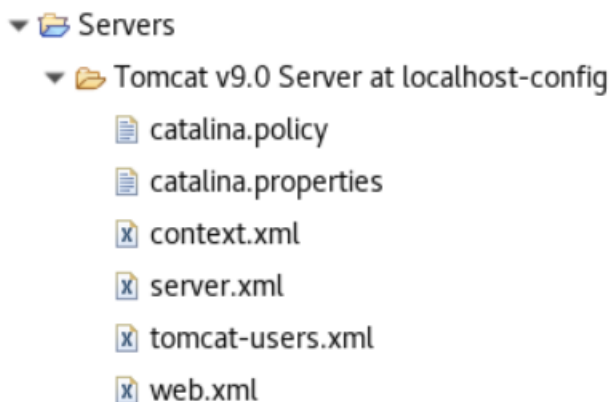
Ein Serverprojekt wird im Workspace erstellt, und ein Tomcat v9.0-Server ist jetzt in der Serveransicht verfügbar. Hier wird die kompilierte Anwendung bereitgestellt und gestartet:



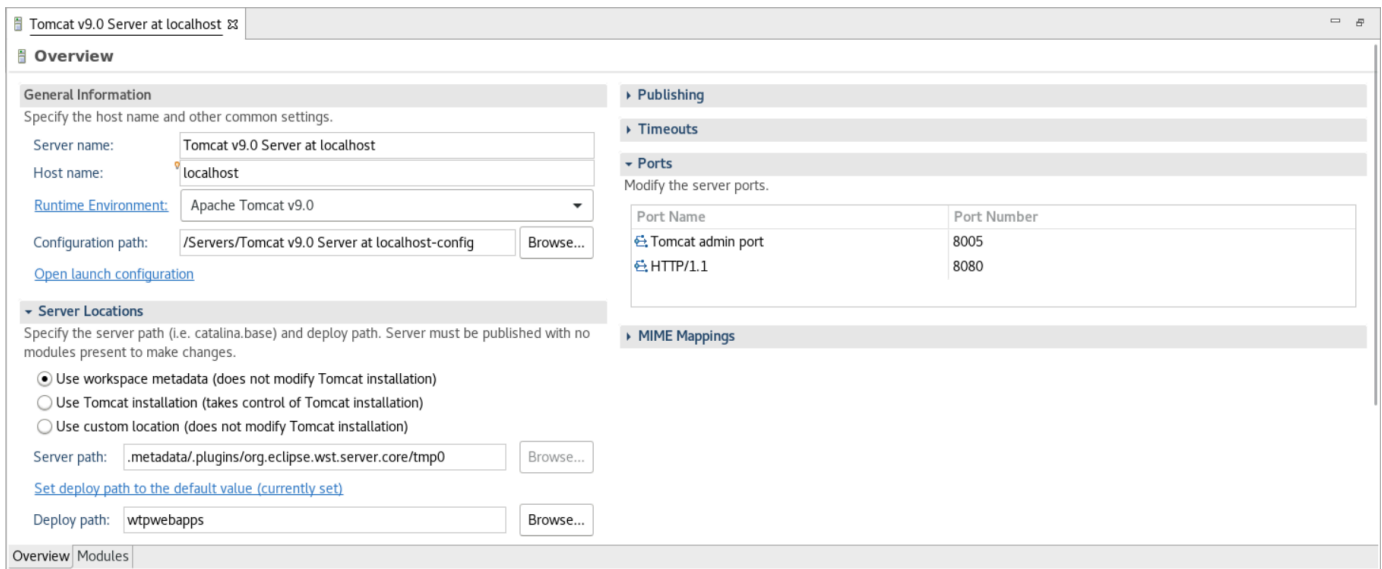
## Schritt 7: Auf Tomcat bereitstellen

In diesem Schritt stellen Sie die Planets-Demo-Anwendung auf dem Tomcat-Server bereit, damit Sie die Anwendung ausführen können.

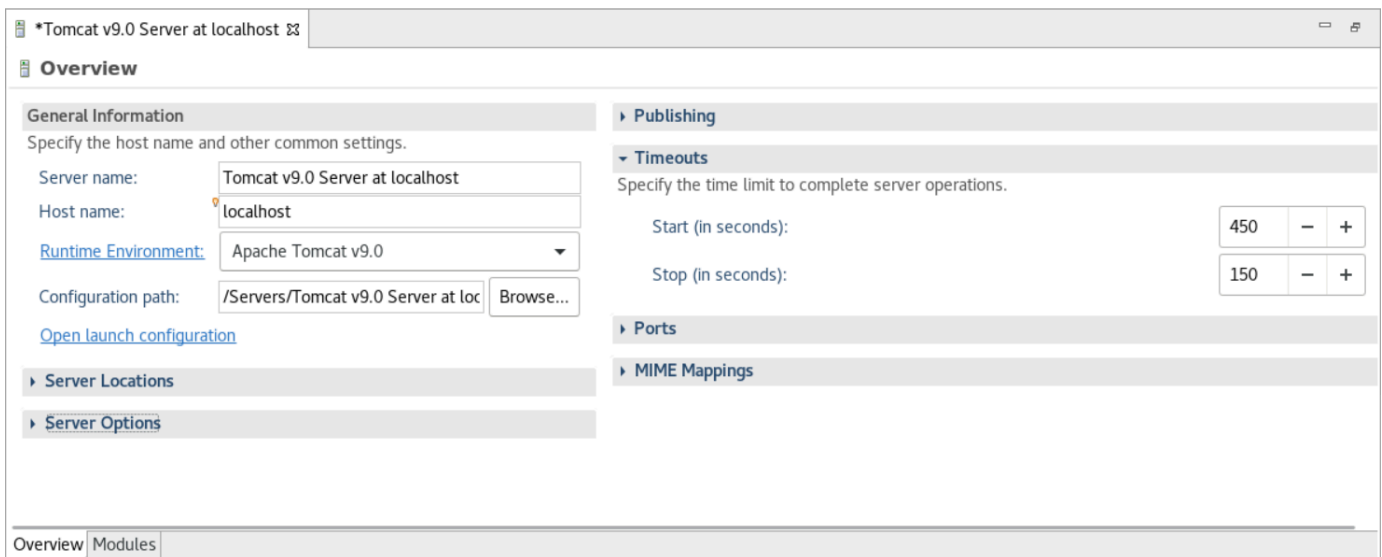
1. Wählen Sie die PlanetsDemo-web Datei aus und wählen Sie „Ausführen als“ > „Maven-Installation“. Wählen Sie PlanetsDemo-web erneut und wählen Sie Refresh, um sicherzustellen, dass das npm-kompilierte Frontend ordnungsgemäß in die Datei .war kompiliert wurde und von Eclipse erkannt wird.
2. Laden Sie die [PlanetsDemoDatei -runtime.zip](#) in die Instanz hoch und entpacken Sie die Datei an einem Ort, auf den Sie zugreifen können. Dadurch wird sichergestellt, dass die Demo-Anwendung auf die benötigten Konfigurationsordner und -dateien zugreifen kann.
3. Kopieren Sie den Inhalt von PlanetsDemo-runtime/tomcat-config in den Servers/Tomcat v9.0... Unterordner, den Sie für Ihren Tomcat-Server erstellt haben:



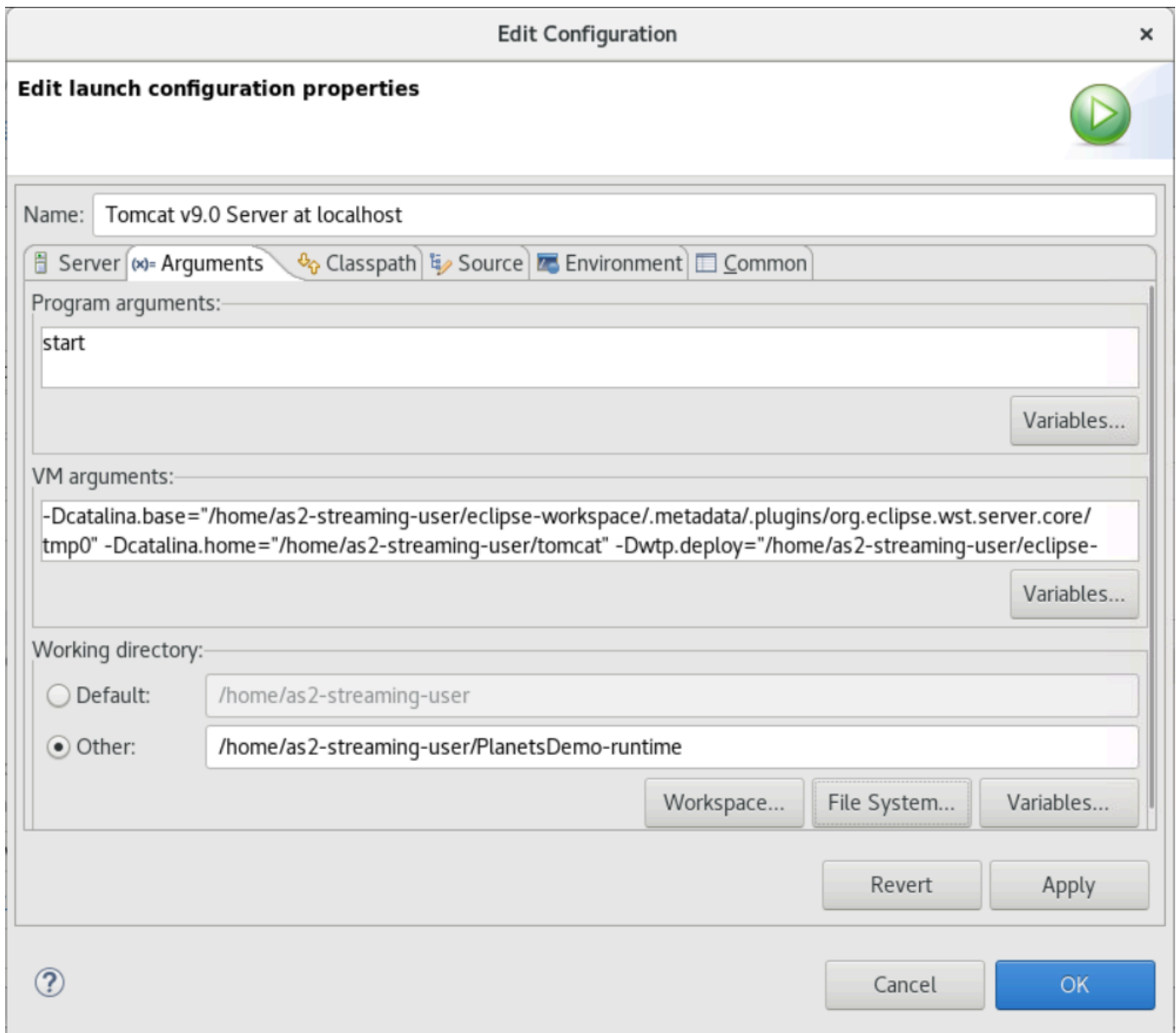
4. Öffnen Sie den tomcat v9.0 Servereintrag in der Serveransicht. Der Servereigenschaften-Editor wird angezeigt:



5. Erhöhen Sie auf der Registerkarte „Übersicht“ die Timeout-Werte auf 450 Sekunden für Start und 150 Sekunden für Stopp, wie hier gezeigt:



6. Wählen Sie Startkonfiguration öffnen. Ein Assistent wird angezeigt. Navigieren Sie im Assistenten zum Ordner Arguments und wählen Sie unter Arbeitsverzeichnis die Option Andere aus. Wählen Sie Dateisystem und navigieren Sie zu dem PlanetsDemo-runtime Ordner, den Sie zuvor entpackt haben. Dieser Ordner sollte einen direkten Unterordner namens config enthalten.

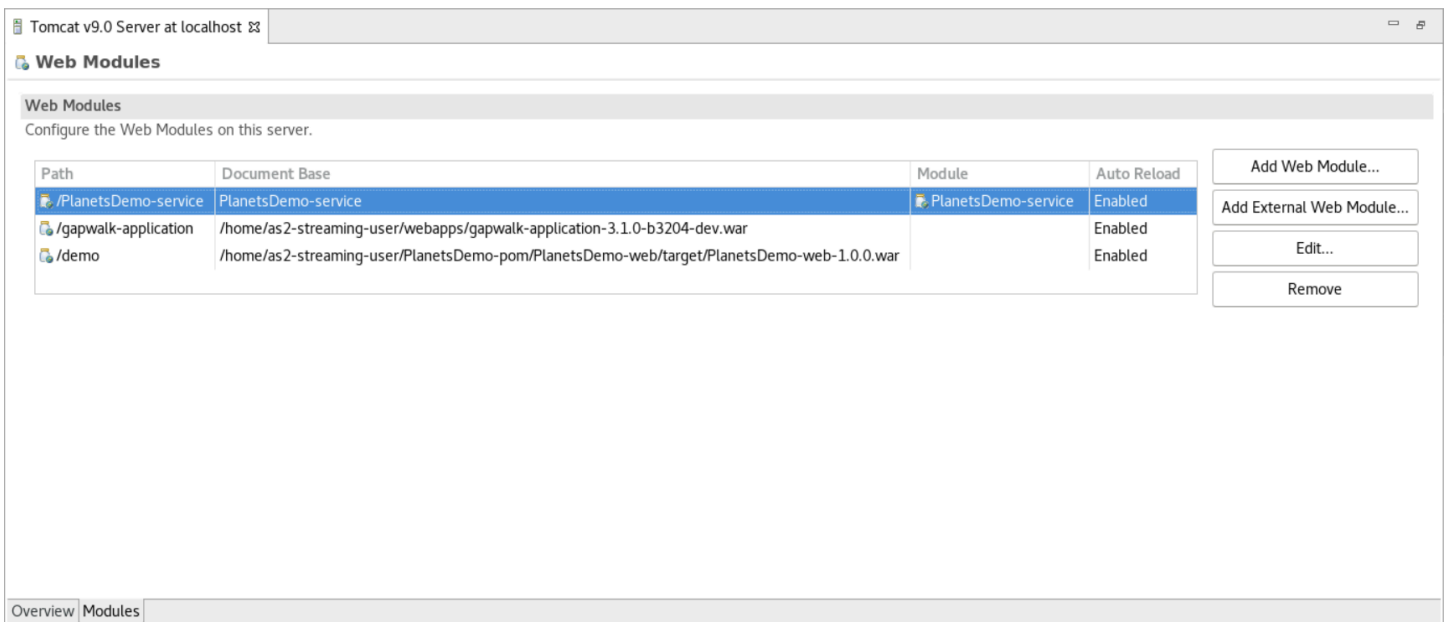


7. Wählen Sie im Servereigenschaften-Editor die Registerkarte Module und nehmen Sie die folgenden Änderungen vor:

- Wählen Sie „Web-Modul hinzufügen“ und „HinzufügenPlanetsDemo-service“.
- Wählen Sie Externes Web-Modul hinzufügen. Das Dialogfenster Web-Modul hinzufügen wird angezeigt. Nehmen Sie die folgenden Änderungen vor:
  - Wählen Sie in Document Base die Option Durchsuchen und navigieren Sie zu `~/webapps/gapwalk-application...war`
  - Geben Sie im Feld Pfad den Wert `ein/gapwalk-application`.
- Wählen Sie OK aus.

- Wählen Sie erneut Externes Web-Modul hinzufügen und nehmen Sie die folgenden Änderungen vor:
  - Geben Sie für Document Base den Pfad zum Frontend .war (in) ein PlanetsDemo-web/target
  - Geben Sie für Pfad Folgendes ein /demo
- Klicken Sie auf OK
- Speichern Sie die Editor-Änderungen (Strg + S).

Der Inhalt des Editors sollte jetzt dem Folgenden ähneln.



## Schritt 8: Erstellen Sie die JICS-Datenbank

In diesem Schritt stellen Sie eine Verbindung zu der Datenbank her, in [Schritt 1: Erstellen einer Datenbank](#) der Sie erstellt haben.

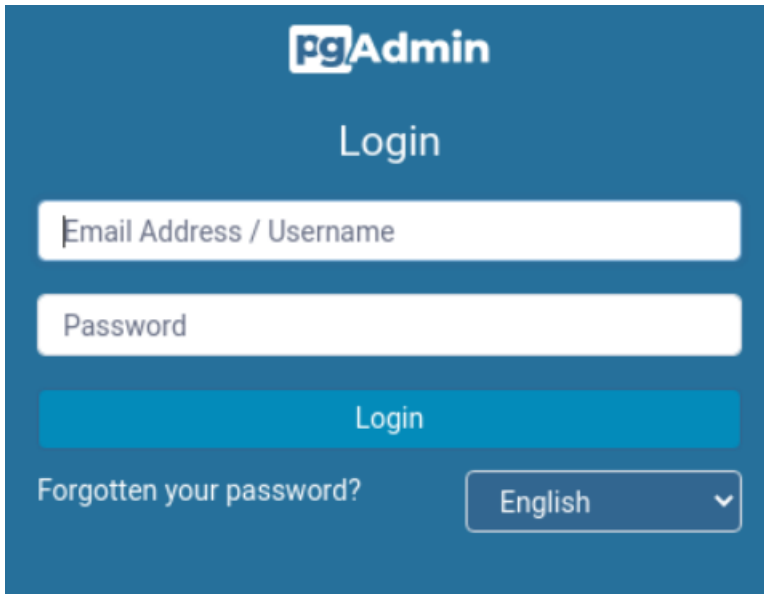
1. Geben Sie in der AppStream 2.0-Instance den folgenden Befehl in einem Terminal ein, um ihn zu startenpgAdmin:

```
./pgadmin-start.sh
```

2. Wählen Sie eine E-Mail-Adresse und ein Passwort als Login-Identifikatoren. Notieren Sie sich die angegebene URL (normalerweise http://127.0.0.1:5050). Starten Sie Google Chrome in der



Instanz, kopieren Sie die URL, fügen Sie sie in den Browser ein und melden Sie sich mit Ihren Identifikatoren an.



3. Nachdem Sie sich angemeldet haben, wählen Sie Neuen Server hinzufügen und geben Sie die Verbindungsinformationen zur zuvor erstellten Datenbank wie folgt ein.

The screenshot shows a configuration window titled "Register - Server" with a "Connection" tab selected. The fields are as follows:

Field	Value
Host name/address	xxx.yyy.zzz.rds.amazonaws.com
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	<input type="checkbox"/>
Password	.....
Save password?	<input type="checkbox"/>
Role	
Service	

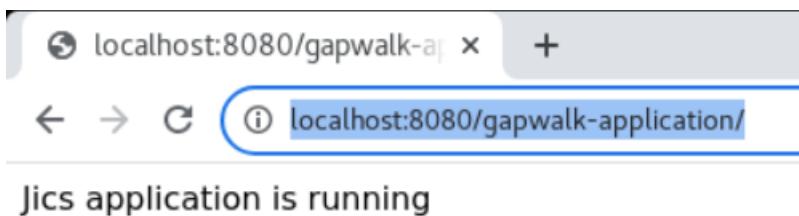
At the bottom, there are buttons for "Close", "Reset", and "Save", along with information and help icons.

4. Wenn Sie eine Verbindung zum Datenbankserver herstellen, verwenden Sie Objekt > Erstellen > Datenbank und erstellen Sie eine neue Datenbank mit dem Namen jics.
5. Bearbeiten Sie die Datenbankverbindungsinformationen, die die Demo-App verwendet hat. Diese Informationen sind in `definiertPlanetsDemo-runtime/config/application-main.yml`. Suchen Sie nach dem `jicsDs` Eintrag. Um die Werte für `username` und `password` in der Amazon RDS-Konsole zu der Datenbank zu navigieren. Wählen Sie auf der Registerkarte Configuration (Konfiguration) unter Master Credentials ARN (Master-Anmeldeinformationen-ARN) die Option Manage in Secrets Manager (In Secrets Manager verwalten) aus. Wählen Sie dann in der Secrets Manager Manager-Konsole im Secret die Option Retrieve Secret Value aus.

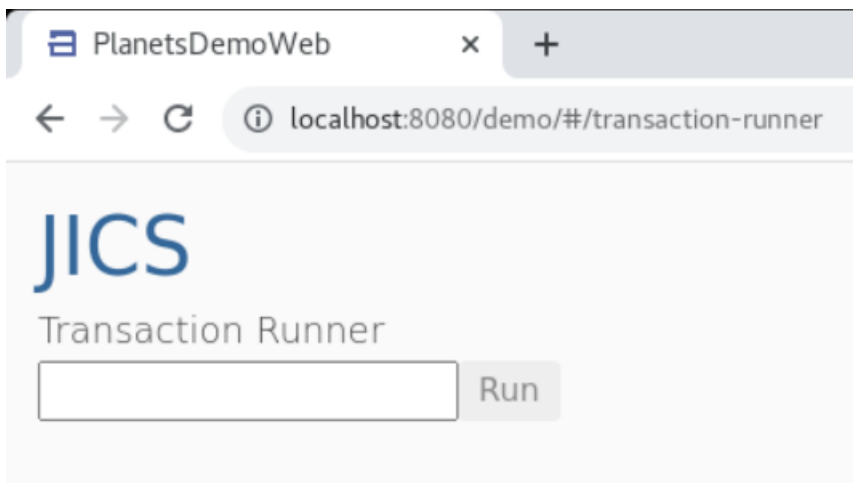
## Schritt 9: Starten und testen Sie die Anwendung

In diesem Schritt starten Sie den Tomcat-Server und die Demo-Anwendung, damit Sie sie testen können.

1. Um den Tomcat-Server und die zuvor bereitgestellten Anwendungen zu starten, wählen Sie den Servereintrag in der Serveransicht aus und wählen Sie Start. Eine Konsole wird angezeigt, in der Startprotokolle angezeigt werden.
2. Überprüfen Sie den Serverstatus in der Serveransicht, oder warten Sie auf die Meldung Serverstart in [xxx] Millisekunden in der Konsole. Überprüfen Sie nach dem Start des Servers, ob die Gapwalk-Anwendung ordnungsgemäß bereitgestellt wurde. Greifen Sie dazu in einem Google Chrome-Browser auf die URL <http://localhost:8080/gapwalk-application/> zu. Sie sollten Folgendes sehen.



3. Greifen Sie über Google Chrome unter <http://localhost:8080/demo> auf das bereitgestellte Anwendungs-Frontend zu. Die folgende Transaction Launcher-Seite sollte angezeigt werden.



4. Um die Anwendungstransaktion zu starten, geben Sie den PINQ Text in das Eingabefeld ein und wählen Sie Ausführen (oder drücken Sie die Eingabetaste).

Der Bildschirm der Demo-App sollte erscheinen.

```
PlanetsDemo-web  Insert Mode  Setup  Theme  Help  Quit

PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

Planet name. . . . . _____

Mass (x10^24kg). . . . . :
Diameter (km). . . . . :
Density (kg/m3). . . . . :
Length of day (h). . . . :
Dist. to sun (x10^6) . . :
Orbital period (days). :
Mean temperature (C) . . :
Number of moons. . . . . :
Has a ring system. . . . :
```

5. Geben Sie einen Planetennamen in das entsprechende Feld ein und drücken Sie die Eingabetaste.

```
PlanetsDemo-web  Insert Mode  Setup  Theme  Help  Quit

PLNMAP1          Planets Data Inquiry          PINQ

Type a planet name, then press Enter.

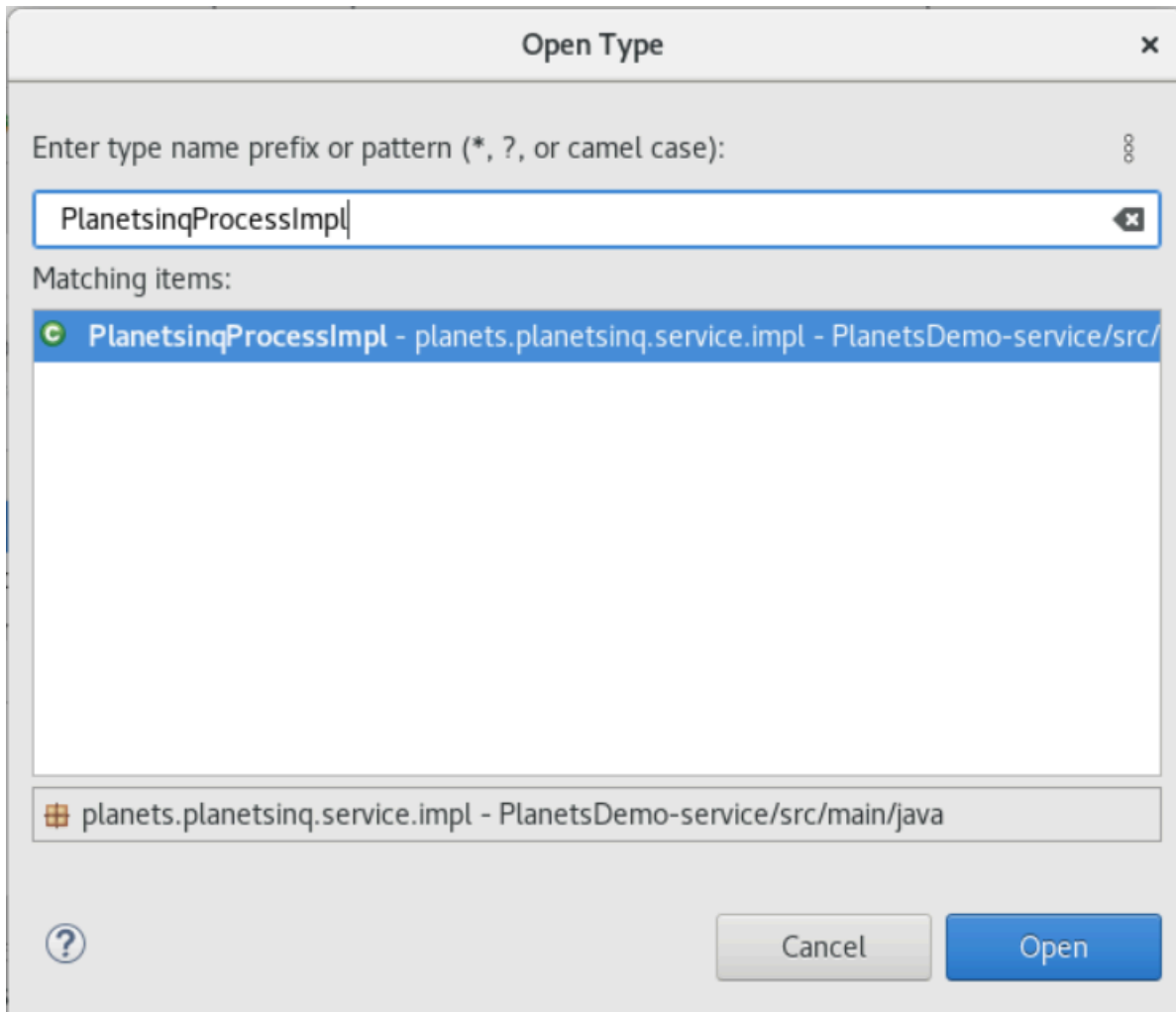
Planet name. . . . . :EARTH

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
```

## Schritt 10: Debuggen Sie die Anwendung

In diesem Schritt testen Sie mit den standardmäßigen Eclipse-Debugging-Funktionen. Diese Funktionen sind verfügbar, wenn Sie an einer modernisierten Anwendung arbeiten.

1. Um die Hauptdienstklasse zu öffnen, drücken Sie Strg + Shift + T. Drücken Sie dann die Eingabetaste. `PlanetsinqProcessImpl`



2. Navigieren Sie zu der `searchPlanet` Methode und fügen Sie dort einen Breakpoint ein.
3. Wählen Sie den Servernamen aus und wählen Sie `In Debug neu starten` aus.
4. Wiederholen Sie die vorherigen Schritte. Rufen Sie also die Anwendung auf, geben Sie einen Planetennamen ein und drücken Sie die Eingabetaste.

Eclipse stoppt die Anwendung in der `searchPlanet` Methode. Jetzt kannst du es untersuchen.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, damit Ihnen keine zusätzlichen Kosten entstehen. Führen Sie folgende Schritte aus:

- Wenn die Planets-Anwendung noch läuft, beenden Sie sie.
- Löschen Sie die Datenbank, in der Sie erstellt haben [Schritt 1: Erstellen einer Datenbank](#). Weitere Informationen finden Sie unter [Löschen einer DB-Instance](#).

# Umgestaltung von Anwendungen mit Micro Focus

In diesem Abschnitt werden die einzelnen Schritte des Replatforming-Prozesses beschrieben. Es beschreibt alle Aufgaben und enthält Informationen zur Konfiguration und zum Betrieb der AWS Mainframe Modernization Runtime auf Amazon EC2.

## Themen

- [Einrichtung von Micro Focus Runtime \(auf Amazon EC2\)](#)
- [Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten](#)
- [Tutorial: Enterprise Analyzer auf AppStream 2.0 einrichten](#)
- [Tutorial: Micro Focus Enterprise Developer auf AppStream 2.0 einrichten](#)
- [Automatisierung für Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer Streaming-Sitzungen einrichten](#)
- [Datensätze als Tabellen und Spalten in Enterprise Developer anzeigen](#)
- [Tutorial: Verwenden von Vorlagen mit Micro Focus Enterprise Developer](#)
- [Tutorial: Den Micro Focus-Build für die BankDemo Beispielanwendung einrichten](#)
- [Tutorial: Einrichtung einer CI/CD-Pipeline zur Verwendung mit Micro Focus Enterprise Developer](#)
- [Batch-Dienstprogramme bei der AWS Mainframe-Modernisierung](#)

## Einrichtung von Micro Focus Runtime (auf Amazon EC2)

AWS Mainframe Modernization bietet mehrere Amazon Machine Images (AMIs), die von Micro Focus lizenzierte Produkte enthalten. Mit diesen AMIs können Sie schnell Amazon Elastic Compute Cloud (Amazon EC2) -Instances bereitstellen, um Micro Focus-Umgebungen zu unterstützen, die Sie kontrollieren und verwalten. In diesem Thema werden die Schritte beschrieben, die für den Zugriff auf und den Start dieser AMIs erforderlich sind. Die Verwendung dieser AMIs ist völlig optional und sie sind nicht erforderlich, um die Tutorials in diesem Benutzerhandbuch abzuschließen.

## Themen

- [Voraussetzungen](#)
- [Erstellen Sie den Amazon VPC-Endpoint für Amazon S3](#)
- [Fordere die Aktualisierung der Zulassungsliste für das Konto an](#)

- [Die AWS Identity and Access Management Rolle wird erstellt](#)
- [Gewähren Sie License Manager die erforderlichen Berechtigungen](#)
- [Abonnieren Sie die Amazon Machine Images](#)
- [Starten Sie eine Micro Focus-Instance für die AWS Mainframe-Modernisierung](#)
- [Subnetz oder VPC ohne Internetzugang](#)
- [Behebung von Lizenzproblemen](#)

## Voraussetzungen

Stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

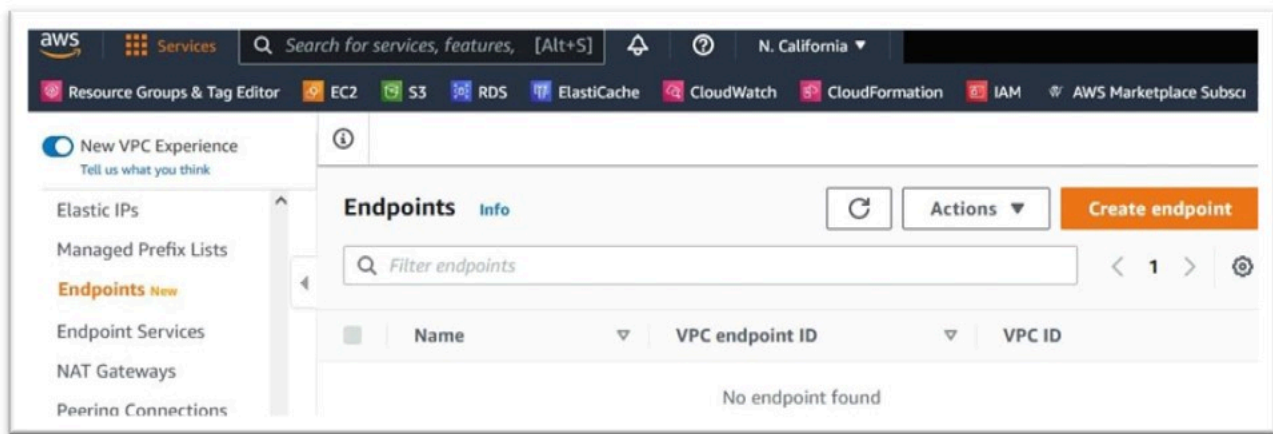
- Administratorzugriff auf das Konto, in dem die Amazon EC2 EC2-Instances erstellt werden.
- Identifizieren Sie den AWS-Region Ort, an dem die Amazon EC2 EC2-Instances erstellt werden, und stellen Sie sicher, dass der AWS Mainframe-Modernisierungsservice verfügbar ist. Weitere Informationen finden Sie unter [AWS Services nach Regionen](#). Achten Sie darauf, eine Region auszuwählen, in der der Service verfügbar ist.
- Identifizieren Sie die Amazon Virtual Private Cloud (Amazon VPC), in der die Amazon EC2 EC2-Instances erstellt werden.

## Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3

In diesem Abschnitt erstellen Sie einen Amazon VPC-Endpunkt, den Amazon S3 verwenden kann.

1. Navigieren Sie zu Amazon VPC in der AWS Management Console.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen.





4. Geben Sie ein aussagekräftiges Namensschild ein, zum Beispiel: „Micro-Focus-License-S3“.
5. Wählen Sie AWS-Services als Servicekategorie.

### Endpoint settings

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

**Service category**  
Select the service category

**AWS services**  
Services provided by Amazon

**PrivateLink Ready partner services**  
Services with an AWS Service Ready designation

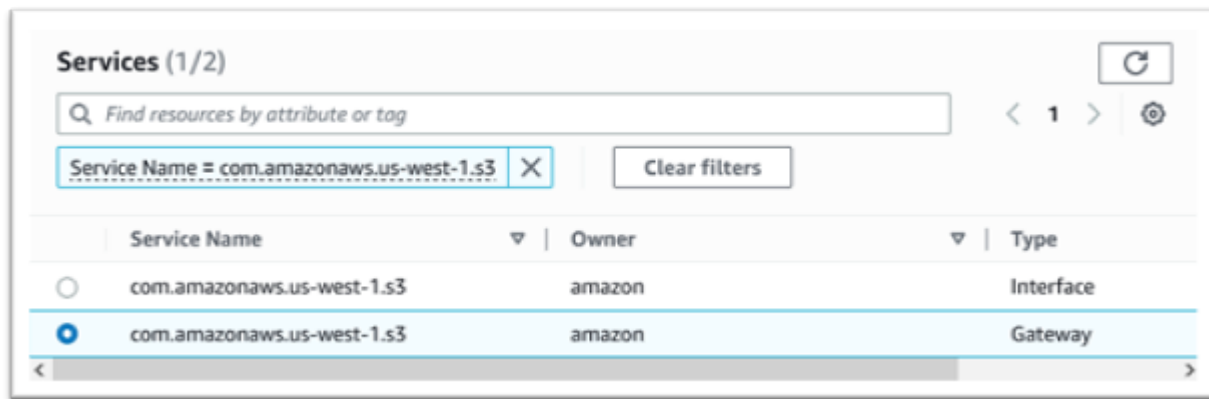
**AWS Marketplace services**  
Services that you've purchased through AWS Marketplace

**Other endpoint services**  
Find services shared with you by service name

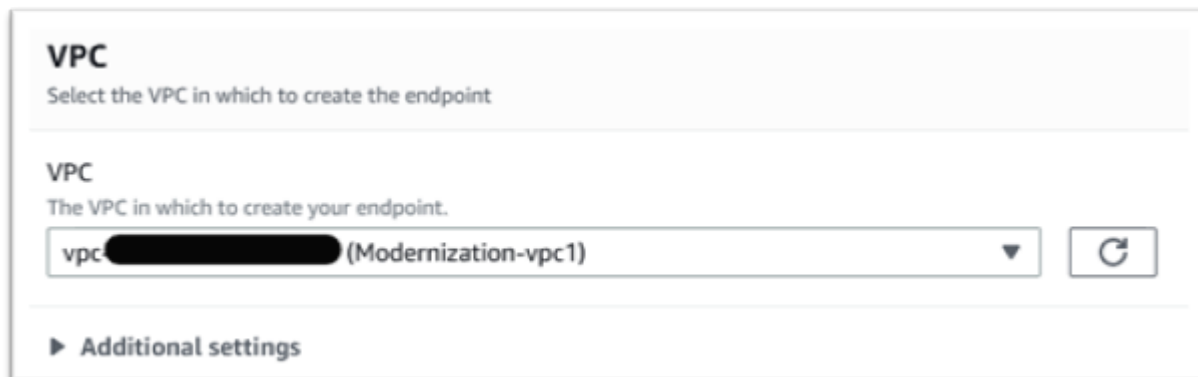
6. Suchen Sie unter Services nach dem Amazon S3 Gateway-Service: com.amazonaws.[Region].s3.

Denn us-west-1 das wäre:com.amazonaws.us-west-1.s3.

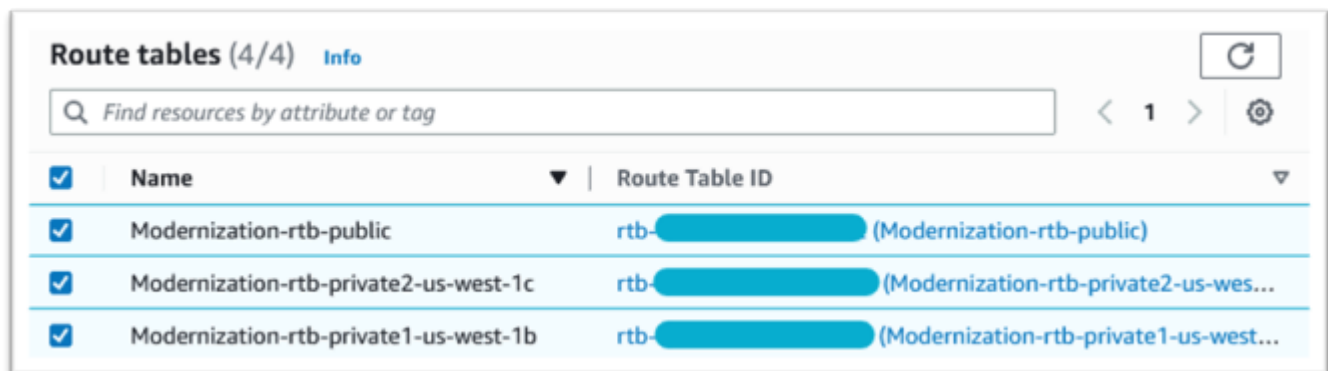
7. Wählen Sie den Gateway-Service.



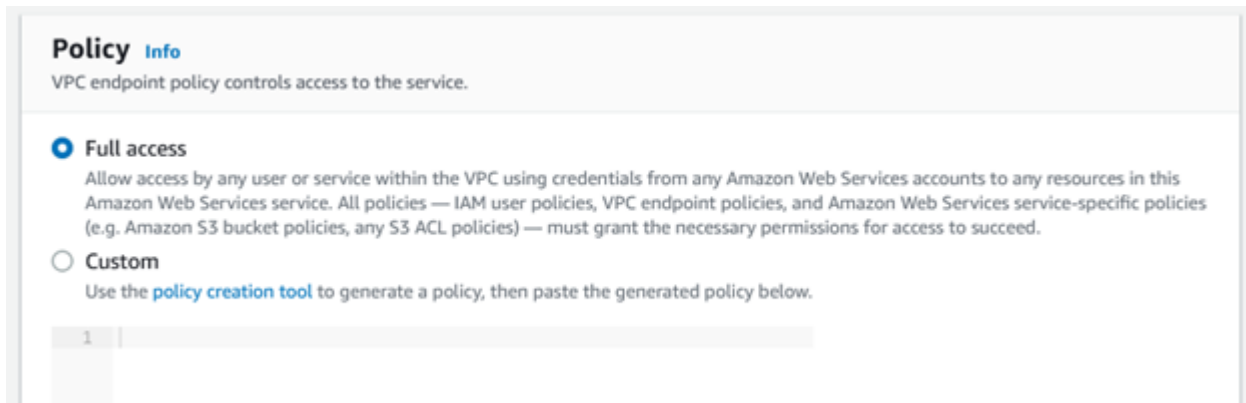
8. Wählen Sie für VPC die VPC aus, die Sie verwenden werden.



9. Wählen Sie alle Routentabellen für die VPC aus.



10. Wählen Sie unter Richtlinie die Option Vollzugriff aus.



11. Klicken Sie auf Endpunkt erstellen.

## Fordere die Aktualisierung der Zulassungsliste für das Konto an

Arbeiten Sie mit Ihrem AWS Kundenbetreuer zusammen, um Ihr Konto für die AWS Mainframe-Modernisierungs-AMIs auf eine Zulassungsliste setzen zu lassen. Bitte geben Sie die folgenden Informationen an:

- Die AWS-Konto ID.
- Der AWS-Region Ort, an dem der Amazon VPC-Endpunkt erstellt wurde.
- Die Amazon VPC Amazon S3 S3-Endpunkt-ID, die in [Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3](#) erstellt wurde. Dies ist die `vpce-xxxxxxxxxxxxxxxxxxxx` ID für die Datei `com.amazonaws.[Region].s3-Gateway-Endpunkt`.
- Die Anzahl der erforderlichen Lizenzen für alle Micro Focus Enterprise Suite AMI Amazon EC2 EC2-Instances.

Eine Lizenz ist pro CPU-Kern erforderlich (pro 2 vCPUs für die meisten Amazon EC2 EC2-Instances).

Weitere Informationen finden Sie unter CPU-Optionen [optimieren](#).

Die angeforderte Nummer kann in future von angepasst werden AWS.

### Note

Der AWS Mitarbeiter muss das Support-Ticket für die Allowlist-Anfrage öffnen. Es kann nicht direkt angefordert werden und die Bearbeitung der Anfrage kann mehrere Tage dauern.

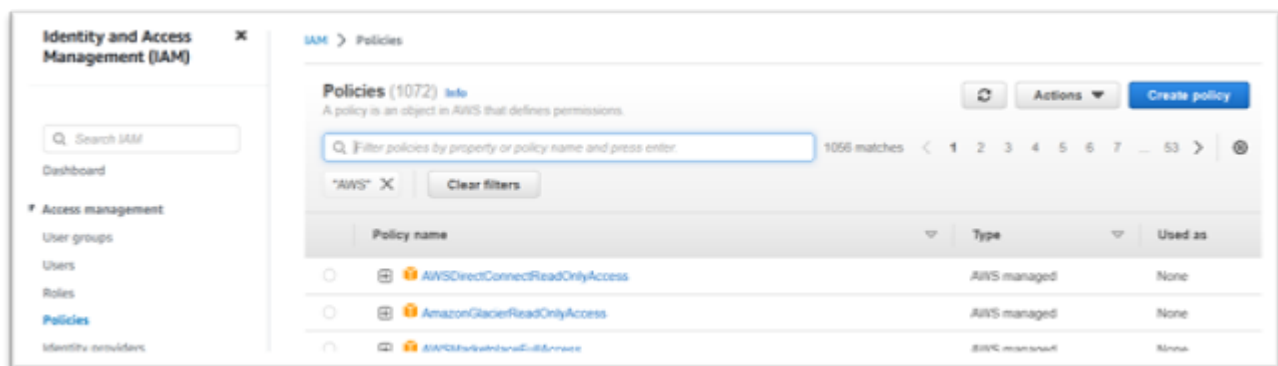
## Die AWS Identity and Access Management Rolle wird erstellt

Erstellen Sie eine AWS Identity and Access Management Richtlinie und Rolle, die von den Amazon EC2 EC2-Instances für die AWS Mainframe-Modernisierung verwendet werden sollen. Wenn Sie die Rolle über die IAM-Konsole erstellen, wird ein zugeordnetes Instance-Profil mit demselben Namen erstellt. Durch die Zuweisung dieses Instance-Profiles zu den Amazon EC2 EC2-Instances können Micro Focus-Lizenzen zugewiesen werden. Weitere Informationen zu Instance-Profilen finden Sie unter [Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf Amazon EC2 EC2-Instances ausgeführt werden](#).

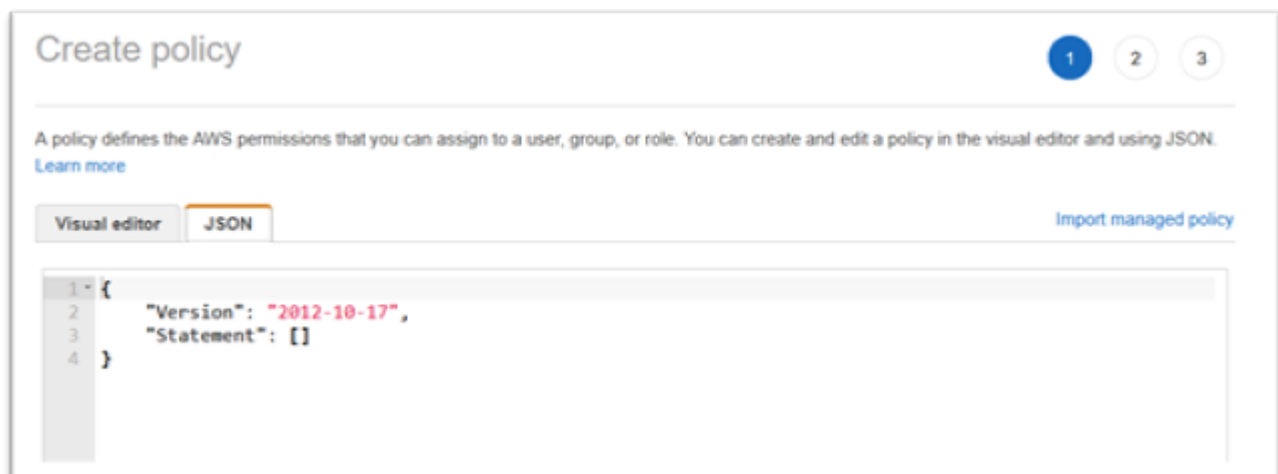
### Erstellen einer IAM-Richtlinie

Eine IAM-Richtlinie wird zuerst erstellt und dann an die Rolle angehängt.

1. Navigieren Sie zu AWS Identity and Access Management in der AWS Management Console.
2. Wählen Sie Richtlinien und dann Richtlinie erstellen aus.



3. Wählen Sie den Tab JSON.



4. Ersetzen Sie `us-west-1` in der folgenden JSON-Datei die AWS-Region Stelle, an der der Amazon S3 S3-Endpoint definiert wurde, und kopieren Sie dann die JSON-Datei und fügen Sie sie in den Richtlinien-Editor ein.

```
{
  "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

Die Aktionen unter der Sid unterstützen `OtherRequiredActions` keine Berechtigungen auf Ressourcenebene und müssen `*` im Ressourcenelement angegeben werden.

1 {  
2 "Version": "2012-10-17",  
3 "Statement": [  
4 {  
5 "Sid": "S3WriteObject",  
6 "Effect": "Allow",  
7 "Action": [  
8 "s3:PutObject"  
9 ],  
10 "Resource": [  
11 "arn:aws:s3::aws-supernova-marketplace-us-west-1-prod/\*"  
12 ]  
13 },  
14 {  
15 "Sid": "OtherRequiredActions",  
16 "Effect": "Allow",  
17 "Action": [  
18 "sts:GetCallerIdentity",  
19 "ec2:DescribeInstances",  
20 "license-manager:ListReceivedLicenses"  
21 ],  
22 "Resource": [  
23 "\*" ]  
24 } ]  
25 }

Character count: 339 of 6,144

5. Wählen Sie Weiter: Markierungen.

**Add tags - optional**  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

**Add tag**  
You can add up to 50 more tags.

6. Geben Sie optional beliebige Tags ein und wählen Sie dann Weiter: Überprüfen.

7. Geben Sie einen Namen für die Richtlinie ein, zum Beispiel „Micro-Focus-Licensing-Policy“. Geben Sie optional eine Beschreibung ein, zum Beispiel „Eine Rolle, die diese Richtlinie enthält, muss jeder Amazon EC2 EC2-Instance für die AWS Mainframe-Modernisierung zugewiesen werden.“

Create policy

1 2 3

### Review policy

**Name\***

Use alphanumeric and '+', '@', '\_' characters. Maximum 128 characters.

**Description**

Maximum 1000 characters. Use alphanumeric and '+', '@', '\_' characters.

**Summary**

Service	Access level	Resource	Request condition
Allow (4 of 369 services) Show remaining 365			
EC2	Limited: List	All resources	None
License Manager	Limited: List	All resources	None
S3	Limited: Write	BucketName   string like   aws-supernova-marketplace-us-west-1-prod, ObjectPath   string like   All	None
STS	Limited: Read	All resources	None

**Tags**

Key	Value
No tags associated with the resource.	

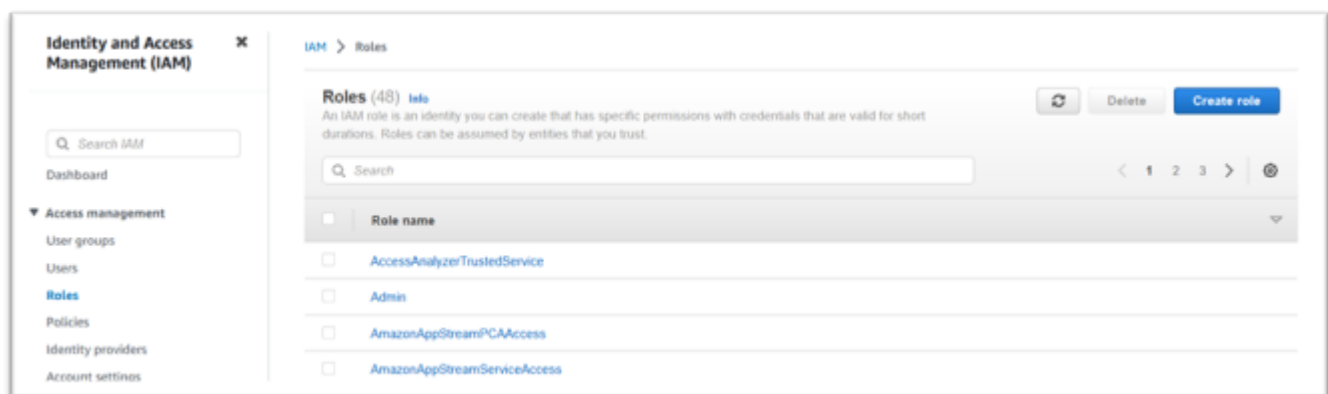
\* Required

Cancel Previous **Create policy**

8. Wählen Sie Richtlinie erstellen aus.

## Erstellen Sie die IAM-Rolle

1. Navigieren Sie zu IAM in der AWS Management Console
2. Wählen Sie Rollen und dann Rolle erstellen aus.

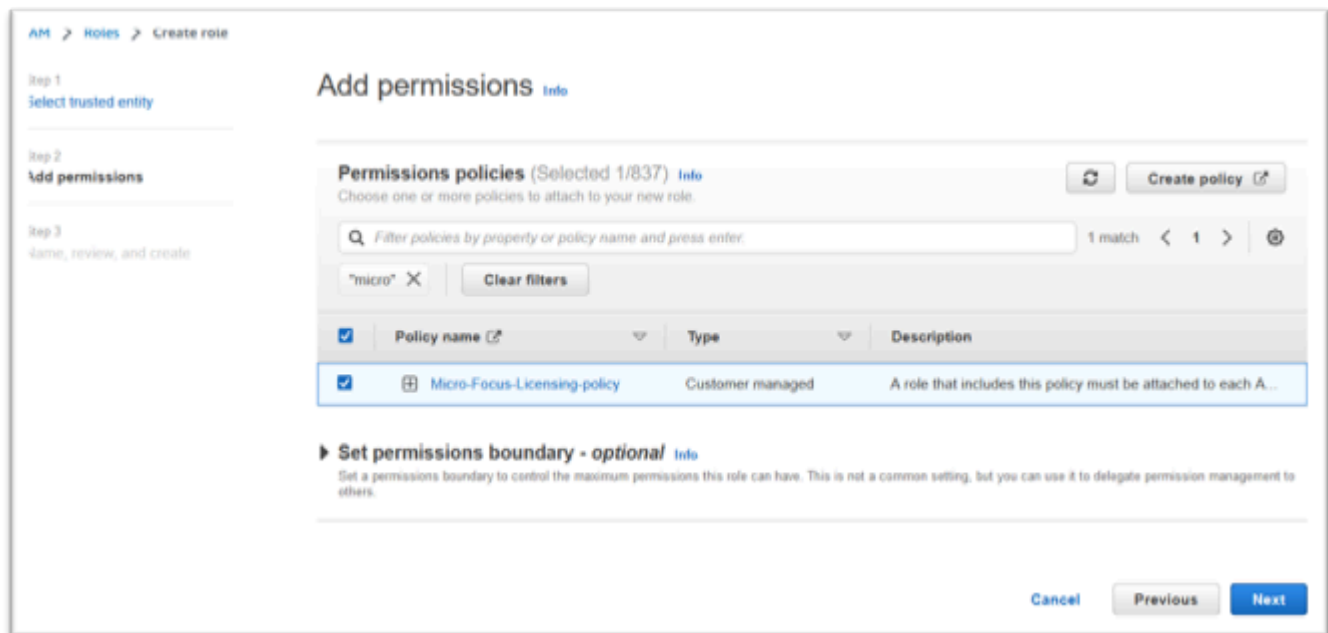


3. Belassen Sie den Entitätstyp Vertrauenswürdig als AWS Dienst und wählen Sie den allgemeinen EC2-Anwendungsfall.

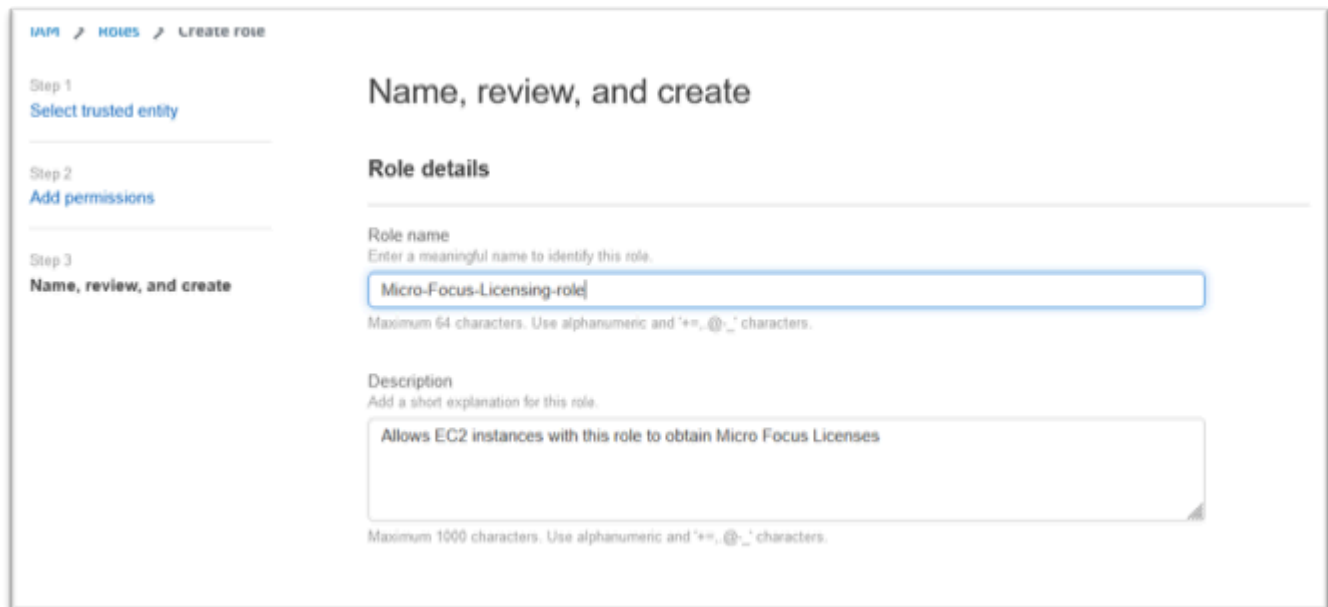
The screenshot shows the 'Select trusted entity' step in the AWS IAM console. The breadcrumb navigation is 'IAM > Roles > Create role'. The left sidebar shows three steps: 'Step 1: Select trusted entity', 'Step 2: Add permissions', and 'Step 3: Name, review, and create'. The main content area is titled 'Select trusted entity' with an 'Info' link. Under 'Trusted entity type', there are five options: 'AWS service' (selected), 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. Each option has a brief description. Below this is the 'Use case' section, which states 'Allow an AWS service like EC2, Lambda, or others to perform actions in this account.' Under 'Common use cases', 'EC2' is selected, with the description 'Allows EC2 instances to call AWS services on your behalf.' 'Lambda' is also listed with the description 'Allows Lambda functions to call AWS services on your behalf.' There is a dropdown menu for 'Use cases for other AWS services' with the placeholder text 'Choose a service to view use case'. At the bottom right, there are 'Cancel' and 'Next' buttons.

4. Wählen Sie Weiter aus.
5. Geben Sie „Micro“ in den Filter ein und drücken Sie die Eingabetaste, um den Filter anzuwenden.
6. Wählen Sie die Richtlinie, die gerade erstellt wurde, zum Beispiel die „Micro-Focus-Licensing-Policy“.
7. Wählen Sie Weiter aus.





8. Geben Sie den Rollennamen ein, zum Beispiel „Micro-Focus-Licensing-Role“.
9. Ersetzen Sie die Beschreibung durch eine eigene, zum Beispiel „Erlaubt Amazon EC2 EC2-Instances mit dieser Rolle, Micro Focus-Lizenzen zu erhalten“.



10. Überprüfen Sie unter Schritt 1: Vertrauenswürdige Entitäten auswählen den JSON-Code und stellen Sie sicher, dass er die folgenden Werte hat:

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

```
"Statement": [  
  {  
    "Effect": "Allow",  
    "Action": [  
      "sts:AssumeRole"  
    ],  
    "Principal": {  
      "Service": [  
        "ec2.amazonaws.com"  
      ]  
    }  
  }  
]
```

**Note**

Die Reihenfolge von „Effekt“, „Aktion“ und „Prinzipal“ ist nicht signifikant.

11. Vergewissern Sie sich, dass in Schritt 2: Berechtigungen hinzufügen Ihre Lizenzierungsrichtlinie angezeigt wird.

Step 2: Add permissions Edit

Permissions policy summary

Policy name <a href="#">↗</a>	Type	Attached as
<a href="#">Micro-Focus-Licensing-policy</a>	Customer managed	Permissions policy

Tags

**Add tags - optional** [Info](#)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

[Add tag](#)

You can add up to 50 more tags.

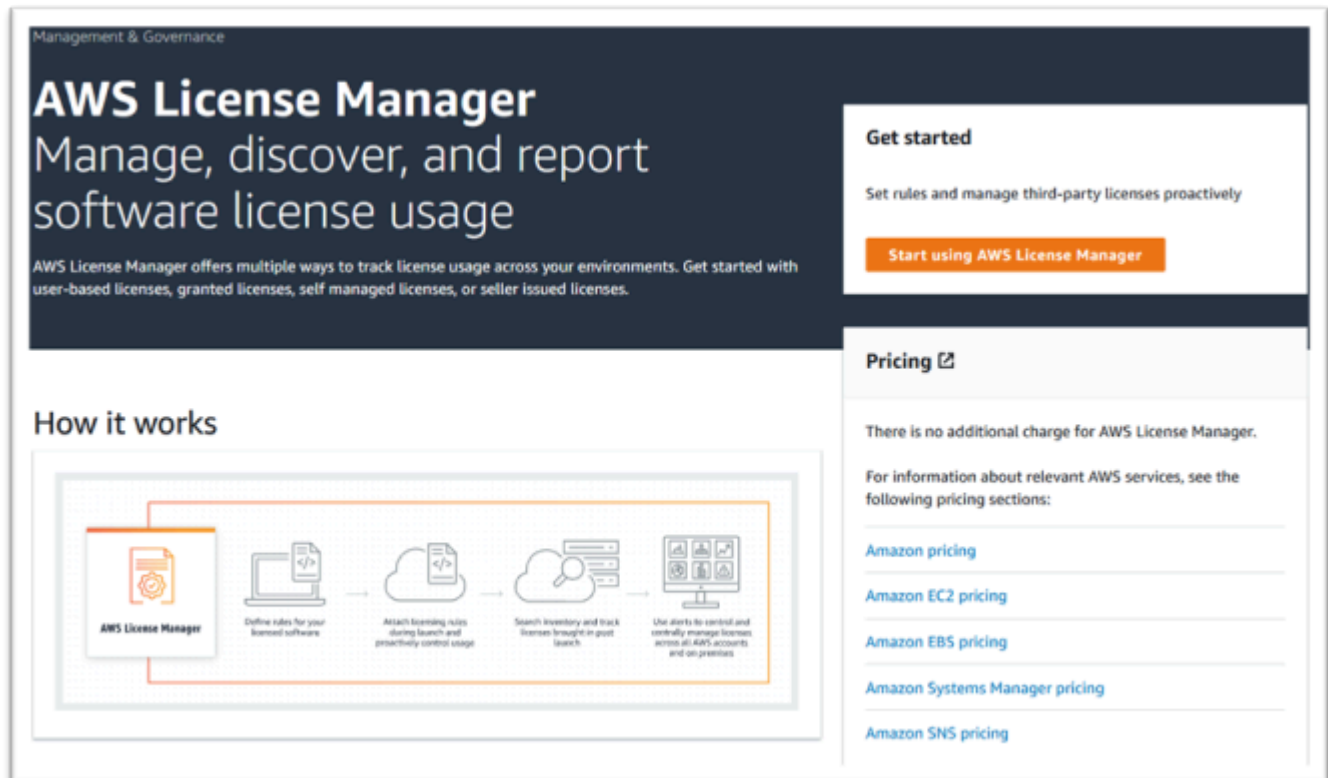
[Cancel](#) [Previous](#) [Create role](#)

12. Wählen Sie Rolle erstellen aus.

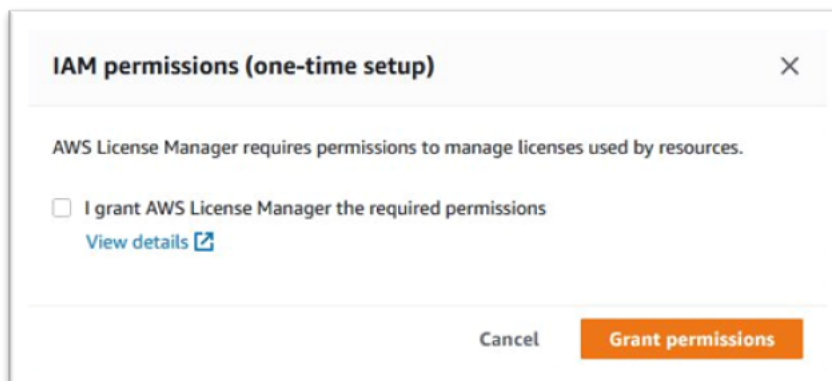
Wenn die Anfrage für die Zulassungsliste abgeschlossen ist, fahren Sie mit den folgenden Schritten fort.

## Gewähren Sie License Manager die erforderlichen Berechtigungen

1. Navigieren Sie zu AWS License Manager in der AWS Management Console.



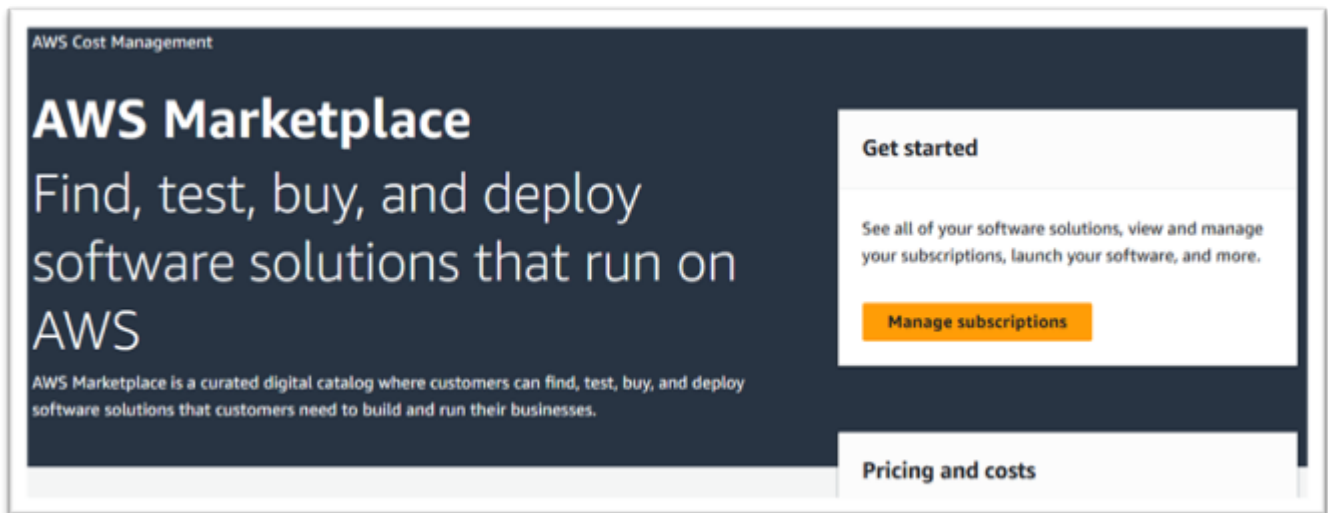
2. Wählen Sie Mit AWS License Manager beginnen aus.
3. Wenn Sie das folgende Pop-up sehen, sehen Sie sich die Details an, aktivieren Sie dann das Kontrollkästchen und klicken Sie auf Berechtigungen gewähren.



## Abonnieren Sie die Amazon Machine Images

Nachdem Sie ein AWS Marketplace Produkt abonniert haben, können Sie eine Instance über das AMI des Produkts starten.

1. Navigieren Sie zu AWS Marketplace Abonnements in der AWS Management Console.
2. Wählen Sie Manage subscriptions (Abonnements verwalten).



3. Kopieren Sie einen der folgenden Links und fügen Sie ihn in die Adressleiste des Browsers ein.


### Note

Wählen Sie nur einen Link für eines der Produkte, zu deren Verwendung Sie autorisiert wurden.

- Unternehmensserver: <https://aws.amazon.com/marketplace/pp/prodview-g5emev63l7blc>
- Enterprise Server für Windows: <https://aws.amazon.com/marketplace/pp/prodview-lwybsiykbhc2>
- Entwickler für Unternehmen: <https://aws.amazon.com/marketplace/pp/prodview-77qmpr42yzxwk>
- Unternehmensentwickler mit Visual Studio 2022: <https://aws.amazon.com/marketplace/pp/prodview-m4l3lqiszo6cm>
- Enterprise Analyser: <https://aws.amazon.com/marketplace/pp/prodview-tttheylcmcihm>

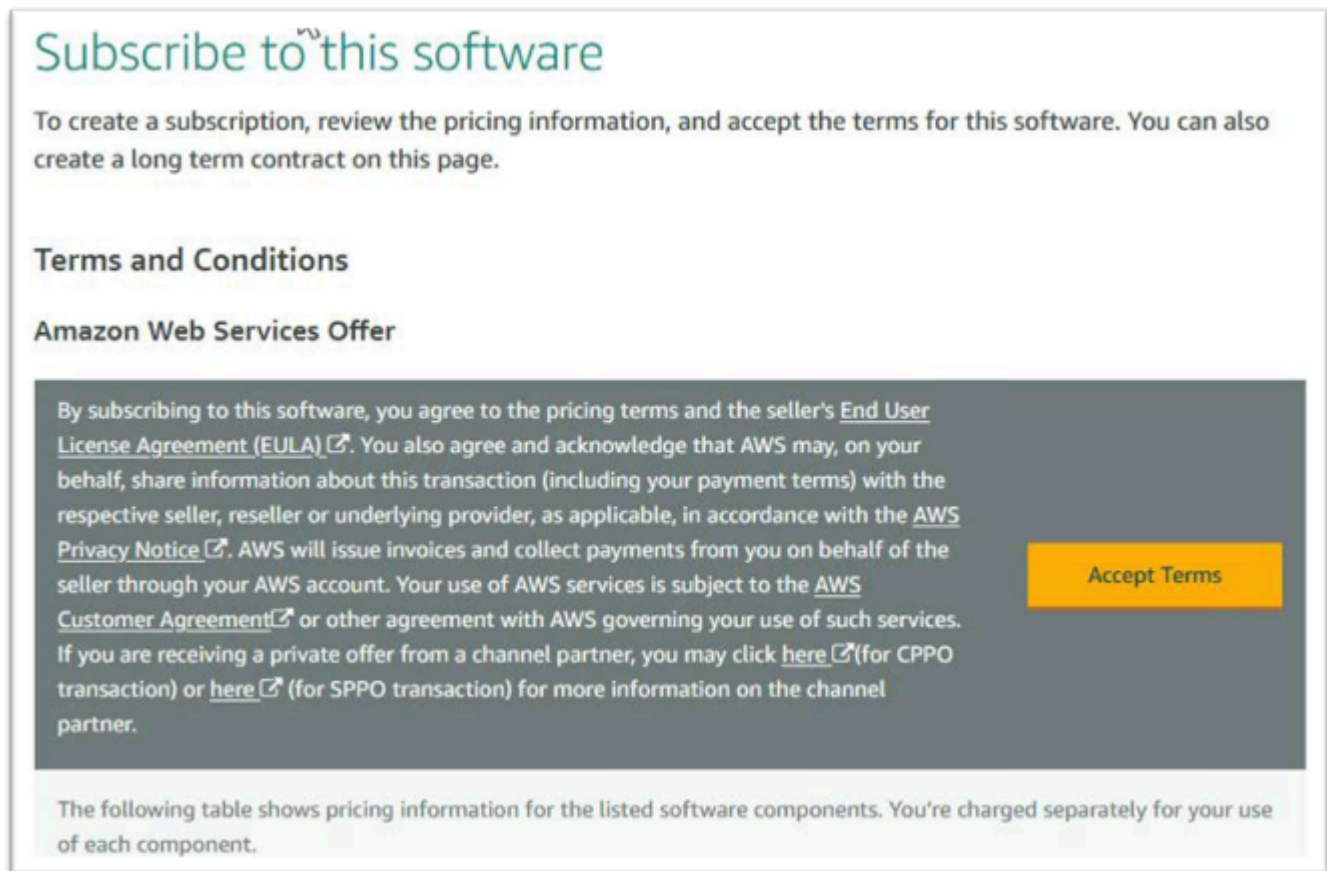
- Enterprise Build Tools für Windows: <https://aws.amazon.com/marketplace/pp/prodview-2rw35bbt6uozl>
- Gespeicherte Unternehmensprozeduren: <https://aws.amazon.com/marketplace/pp/prodview-zoeyqnsdsj6ha>
- Gespeicherte Unternehmensprozeduren mit SQL Server 2019: <https://aws.amazon.com/marketplace/pp/prodview-ynfklquwubnz4>

4. Wählen Sie Continue to Subscribe (Weiter abonnieren) aus.



The screenshot shows the AWS Marketplace listing for Micro Focus Enterprise Server. The listing includes the Micro Focus logo, the product name "Enterprise Server", and the text "By: Amazon Web Services" and "Latest Version: 8.0.1". A description states: "Micro Focus Enterprise Server is a mainframe-compatible deployment environment for COBOL and PL/I applications. Linux/Unix". On the right side, there is a yellow "Continue to Subscribe" button, a "Save to List" button, and a pricing box showing "Typical Total Price \$11.292/hr" with a note: "Total pricing per instance for services hosted on m6i.xlarge in US East (N. Virginia). View Details". At the bottom, there are navigation tabs for "Overview", "Pricing", "Usage", "Support", and "Reviews".

5. Wenn die Allgemeinen Geschäftsbedingungen akzeptabel sind, wählen Sie Bedingungen akzeptieren.



**Subscribe to this software**

To create a subscription, review the pricing information, and accept the terms for this software. You can also create a long term contract on this page.

### Terms and Conditions

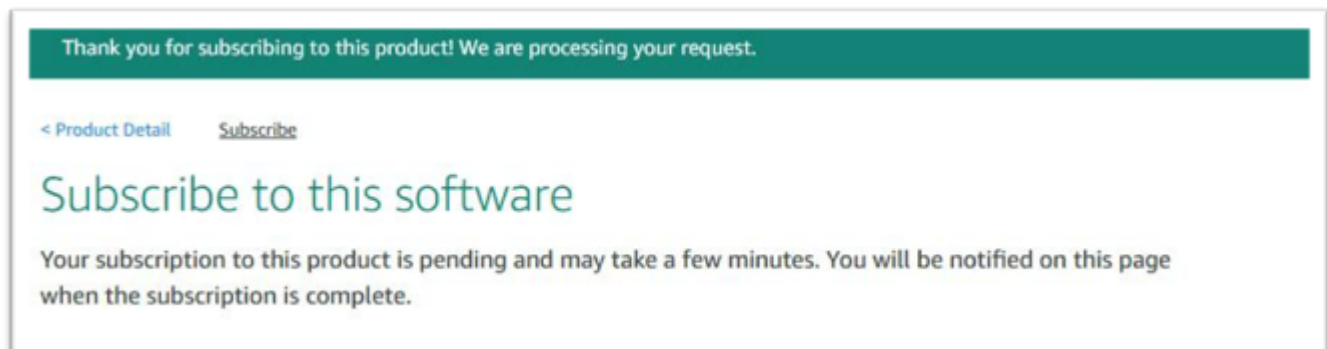
#### Amazon Web Services Offer

By subscribing to this software, you agree to the pricing terms and the seller's [End User License Agreement \(EULA\)](#). You also agree and acknowledge that AWS may, on your behalf, share information about this transaction (including your payment terms) with the respective seller, reseller or underlying provider, as applicable, in accordance with the [AWS Privacy Notice](#). AWS will issue invoices and collect payments from you on behalf of the seller through your AWS account. Your use of AWS services is subject to the [AWS Customer Agreement](#) or other agreement with AWS governing your use of such services. If you are receiving a private offer from a channel partner, you may click [here](#) (for CPPO transaction) or [here](#) (for SPPO transaction) for more information on the channel partner.

**Accept Terms**

The following table shows pricing information for the listed software components. You're charged separately for your use of each component.

- Die Bearbeitung des Abonnements kann einige Minuten dauern.



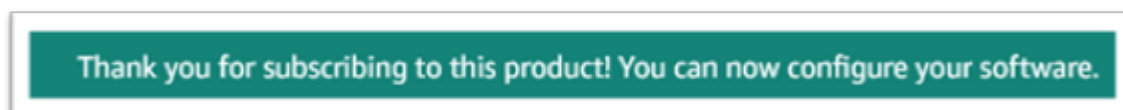
Thank you for subscribing to this product! We are processing your request.

[< Product Detail](#)   [Subscribe](#)

### Subscribe to this software

Your subscription to this product is pending and may take a few minutes. You will be notified on this page when the subscription is complete.

- Nachdem die Dankesnachricht angezeigt wird, kopieren Sie den nächsten Link aus Schritt 3 und fügen Sie ihn ein, um weitere Abonnements hinzuzufügen.

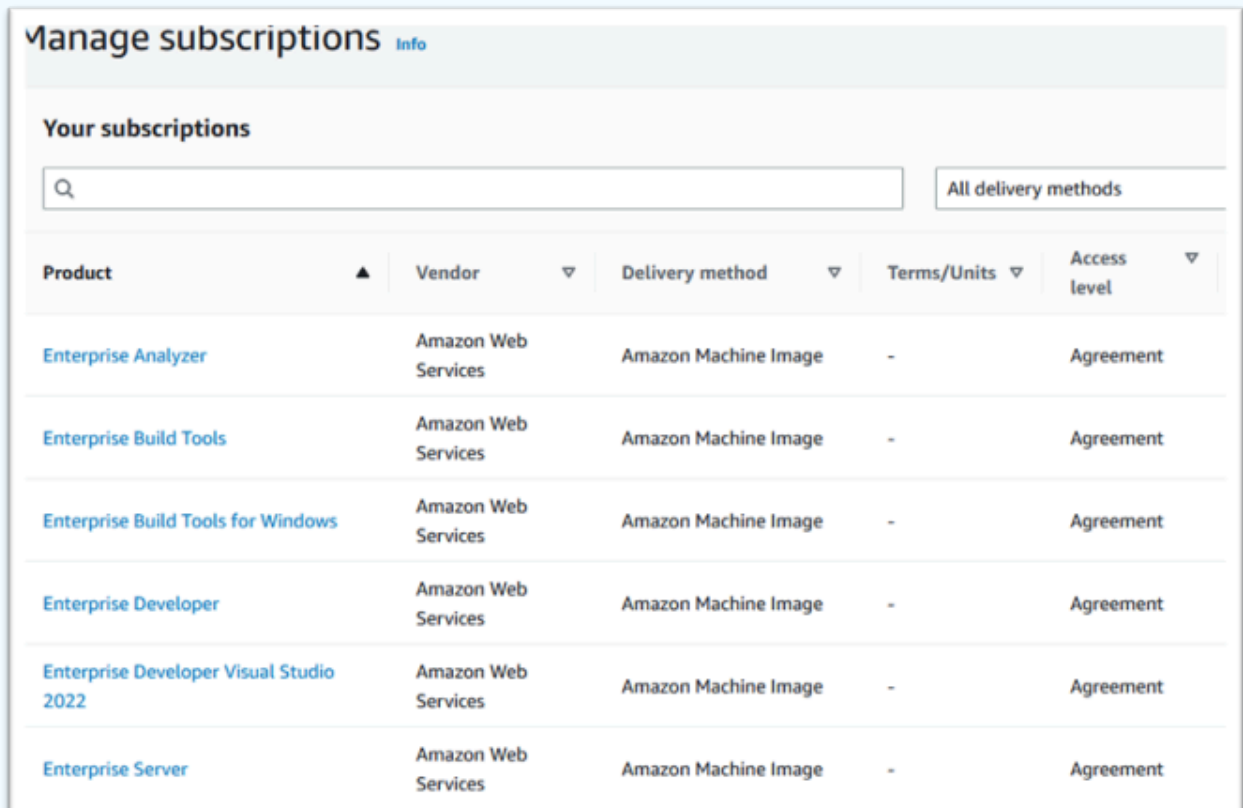


Thank you for subscribing to this product! You can now configure your software.

- Beenden Sie, wenn „Abonnements verwalten“ all Ihre abonnierten AMIs anzeigt.

**Note**

Die Panel-Einstellungen (Zahnradssymbol) sind so eingestellt, dass die Ansicht als Tabelle angezeigt wird.

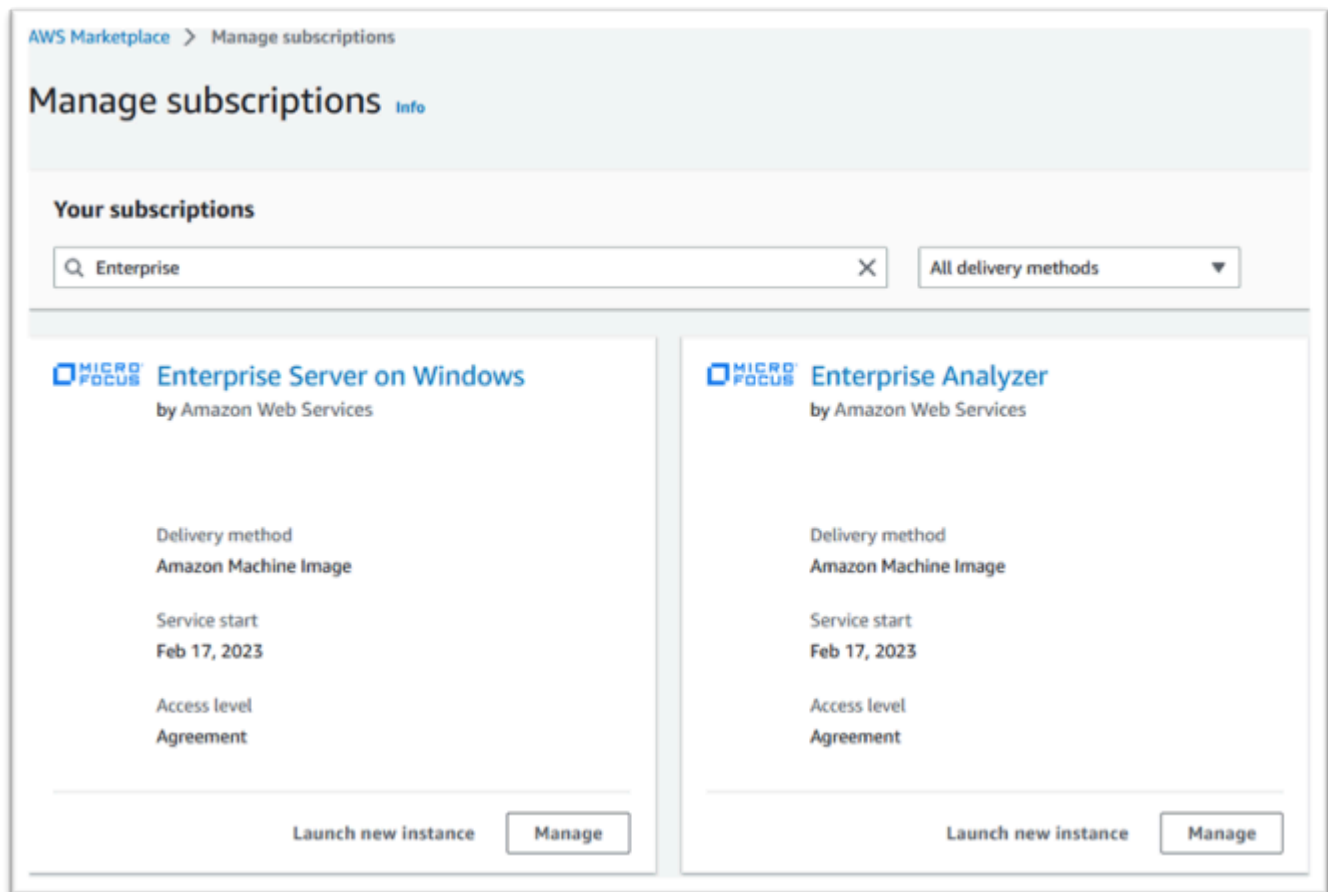


The screenshot shows the 'Manage subscriptions' interface in the AWS Marketplace. It features a search bar and a dropdown for 'All delivery methods'. Below is a table with columns for Product, Vendor, Delivery method, Terms/Units, and Access level. The table lists several subscriptions from Amazon Web Services, all using Amazon Machine Image as the delivery method and having an 'Agreement' access level.

Product	Vendor	Delivery method	Terms/Units	Access level
Enterprise Analyzer	Amazon Web Services	Amazon Machine Image	-	Agreement
Enterprise Build Tools	Amazon Web Services	Amazon Machine Image	-	Agreement
Enterprise Build Tools for Windows	Amazon Web Services	Amazon Machine Image	-	Agreement
Enterprise Developer	Amazon Web Services	Amazon Machine Image	-	Agreement
Enterprise Developer Visual Studio 2022	Amazon Web Services	Amazon Machine Image	-	Agreement
Enterprise Server	Amazon Web Services	Amazon Machine Image	-	Agreement

## Starten Sie eine Micro Focus-Instance für die AWS Mainframe-Modernisierung

1. Navigieren Sie zu AWS Marketplace Abonnements in der AWS Management Console.
2. Suchen Sie das AMI, das gestartet werden soll, und wählen Sie Neue Instanz starten aus.



3. Stellen Sie im Dialogfeld „Neue Instanz starten“ sicher, dass die Region auf der Zulassungsliste ausgewählt ist.
4. Drücken Sie auf Weiter, um über EC2 zu starten.

**Note**

Das folgende Beispiel zeigt den Start eines Enterprise Developer AMI, aber der Prozess ist für alle AWS Mainframe-Modernisierungs-AMIs derselbe.



AWS Marketplace > Manage subscriptions > Enterprise Developer > Launch new instance

## Launch new instance

**Configure this software**  
Choose a fulfillment option below to select how you wish to deploy the software, then enter the information required to configure the deployment.

**Delivery method**  
64-bit (x86) Amazon Machine Image ▼

**Software version**  
v8.0.1 (Oct 26, 2022) ▼  
For older software versions, please visit the [full AWS Marketplace website](#) .

**Region**  
us-west-1 ▼

**AMI ID:** ami-0f199167bc5fce009

Cancel **Continue to launch through EC2**

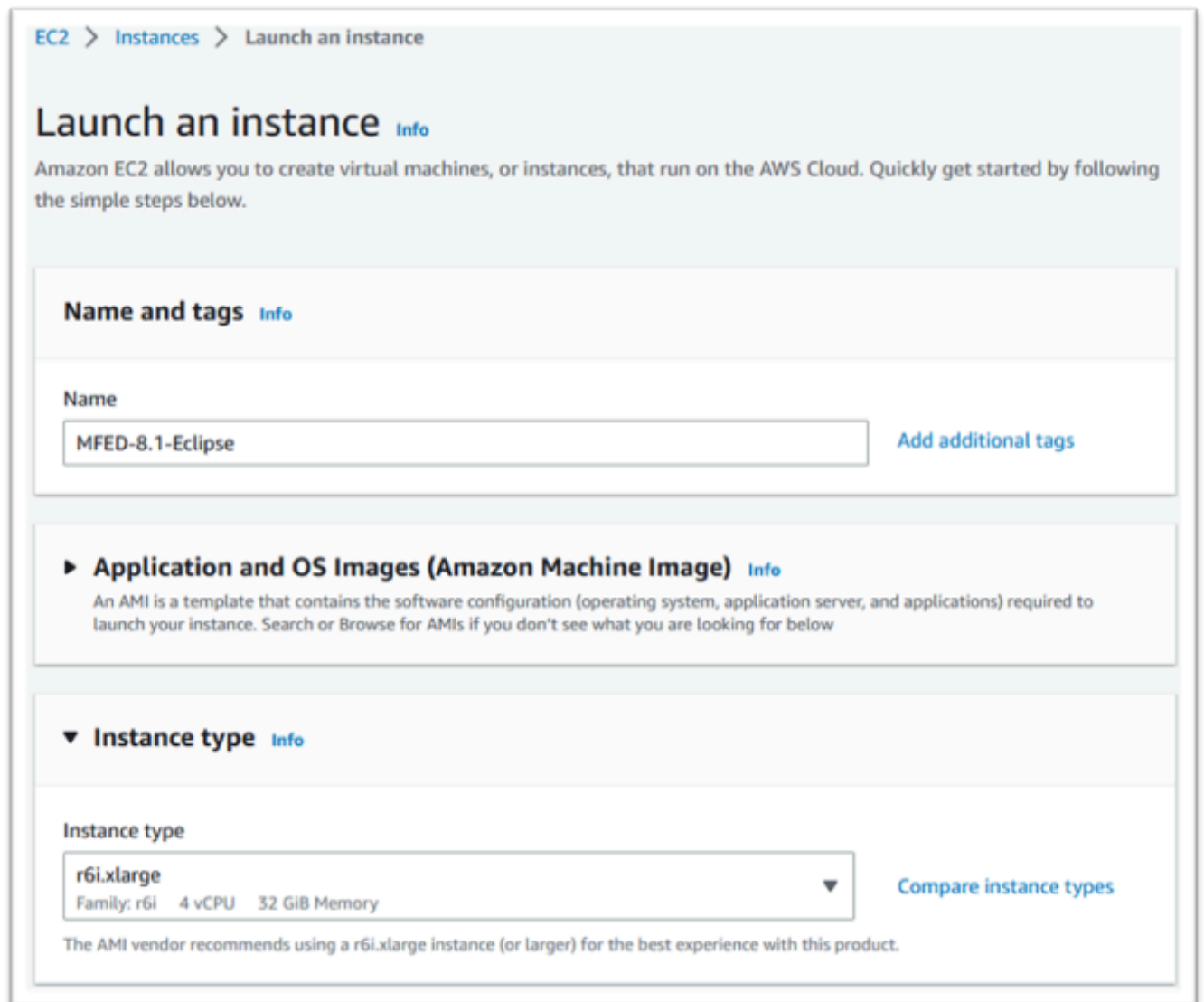
5. Geben Sie einen Namen für den Server ein.
6. Wählen Sie einen Instance-Type.

Der gewählte Instanztyp sollte sich nach den Leistungs- und Kostenanforderungen des Projekts richten. Im Folgenden werden Ausgangspunkte vorgeschlagen:

- Für Enterprise Analyzer ein r6i.xlarge
- Für Enterprise Developer ein r6i.large
- Für eine eigenständige Instanz von Enterprise Server ein r6i.xlarge
- Für Micro Focus Performance Availability Cluster (PAC) mit Scale-Out ein r6i.large

**Note**

Der Bereich Anwendungs- und Betriebssystem-Images wurde für den Screenshot ausgeblendet.



EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags Info

Name

 [Add additional tags](#)

### ▶ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

### ▼ Instance type Info

Instance type

 [Compare instance types](#)  
Family: r6i 4 vCPU 32 GiB Memory

The AMI vendor recommends using a r6i.xlarge instance (or larger) for the best experience with this product.


7. Wählen oder erstellen (und speichern) Sie ein Schlüsselpaar (nicht angezeigt).

Weitere Informationen zu Schlüsselpaaren für Linux-Instances finden Sie unter [Amazon EC2 EC2-Schlüsselpaare und Linux-Instances](#).

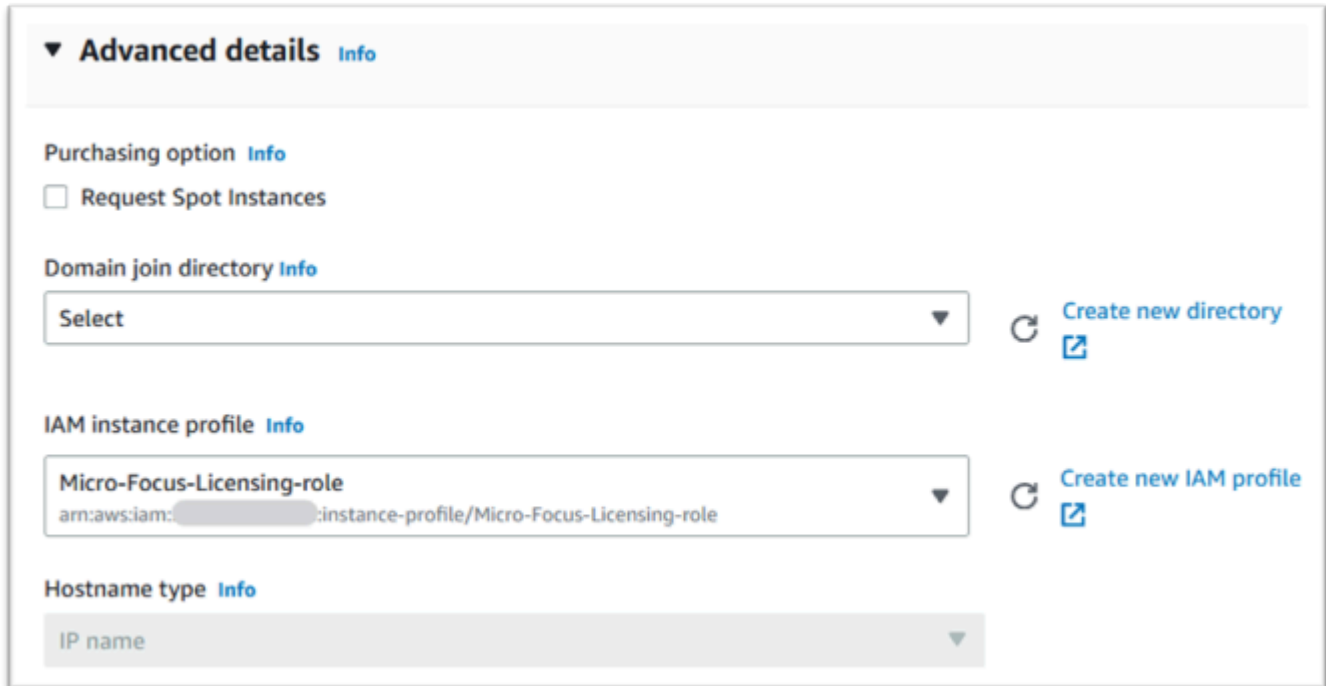
Weitere Informationen zu Schlüsselpaaren für Windows-Instances finden Sie unter [Amazon EC2 EC2-Schlüsselpaare und Windows-Instances](#).

8. Bearbeiten Sie die Netzwerkeinstellungen und wählen Sie die zugelassene VPC und das entsprechende Subnetz aus.

9. Wählen oder erstellen Sie eine Sicherheitsgruppe. Wenn es sich um eine Enterprise Server EC2-Instance handelt, ist es üblich, TCP-Verkehr zu den Ports 86 und 10086 zuzulassen, um die Micro Focus-Konfiguration zu verwalten.
10. Konfigurieren Sie optional den Speicher für die Amazon EC2 EC2-Instance.
11. Wichtig — Erweitern Sie Erweiterte Details und wählen Sie unter IAM-Instance-Profil die zuvor erstellte Lizenzierungsrolle aus, zum Beispiel „Micro-Focus-Licensing-Role“.

 Note

Wenn dieser Schritt versäumt wird, können Sie nach der Erstellung der Instance die IAM-Rolle über die Option Sicherheit des Aktionsmenüs für die EC2-Instance ändern.



**Advanced details** [Info](#)

**Purchasing option** [Info](#)

Request Spot Instances

**Domain join directory** [Info](#)

Select [Create new directory](#)

**IAM instance profile** [Info](#)

Micro-Focus-Licensing-role  
arn:aws:iam:[:redacted]:instance-profile/Micro-Focus-Licensing-role [Create new IAM profile](#)

**Hostname type** [Info](#)

IP name

12. Sehen Sie sich die Zusammenfassung an und starten Sie die Instance per Push.

▼ **Summary**

Number of instances [Info](#)

1

Software Image (AMI)

Distribution Configuration for...[read more](#)  
ami-0f199167bc5fce009

Virtual server type (instance type)

r6i.xlarge

Firewall (security group)

default

Storage (volumes)

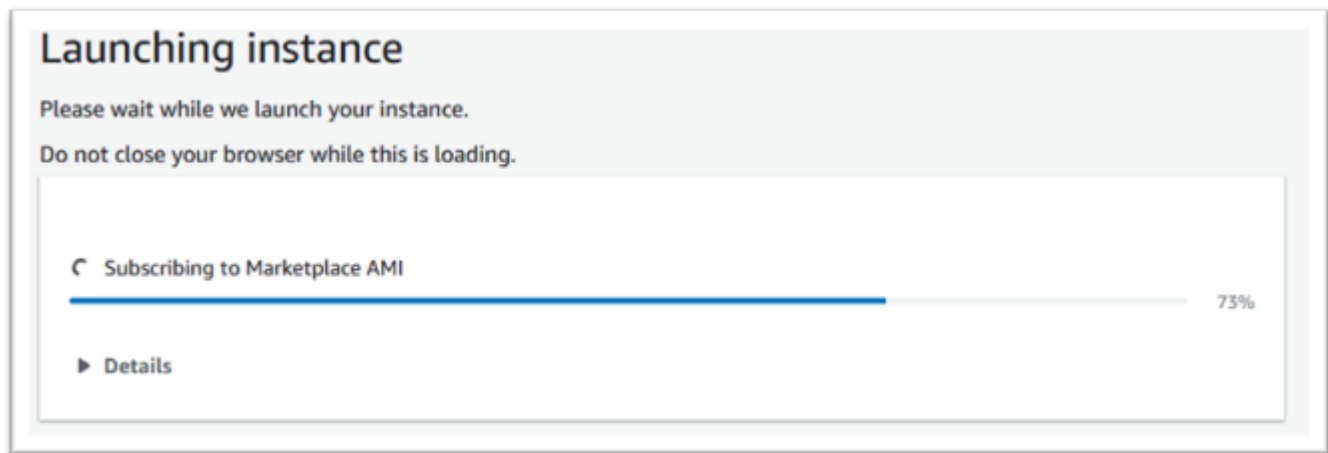
1 volume(s) - 100 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

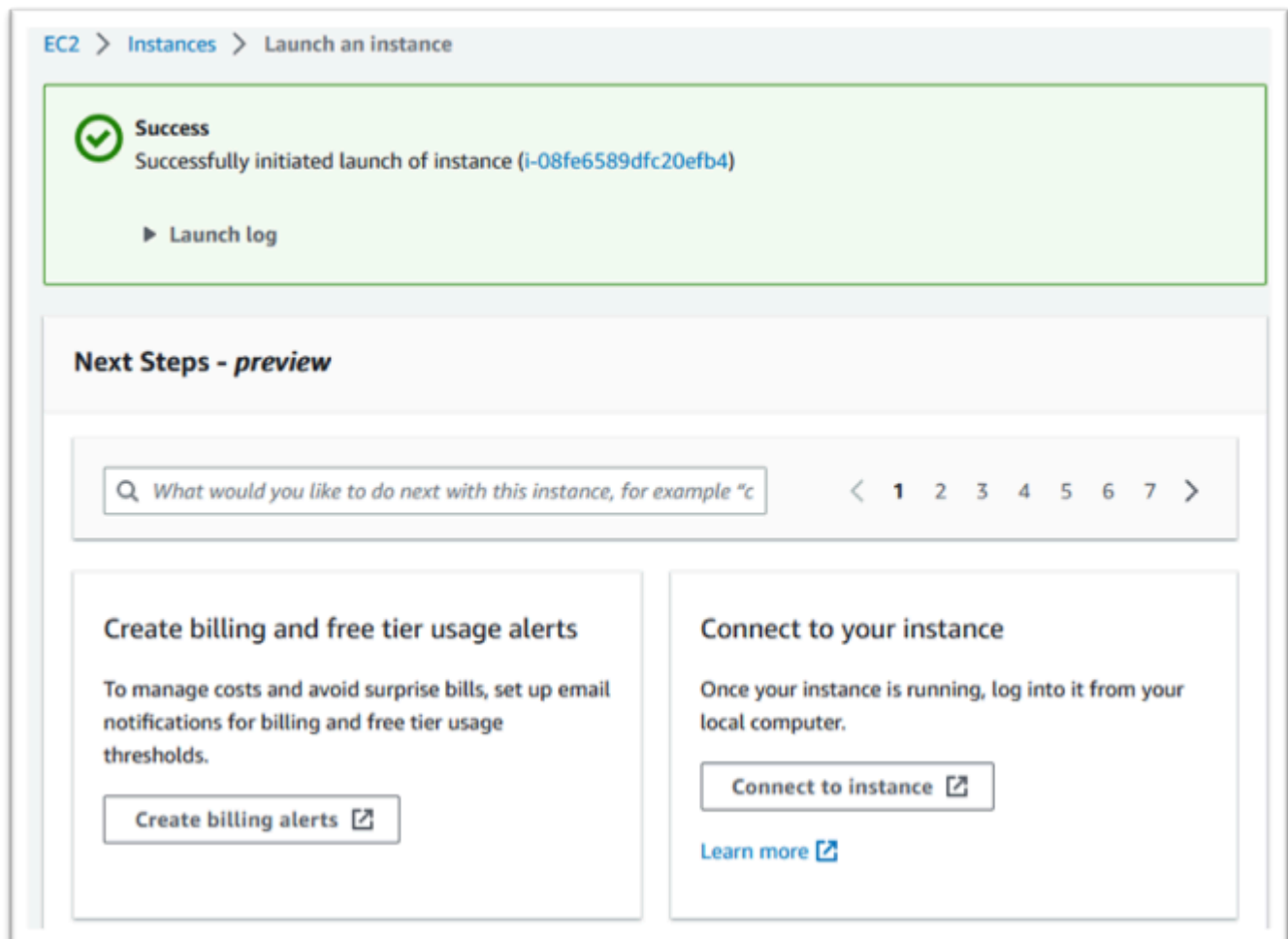
Cancel Launch instance

13. Der Instanzstart schlägt fehl, wenn ein ungültiger virtueller Servertyp ausgewählt wird.

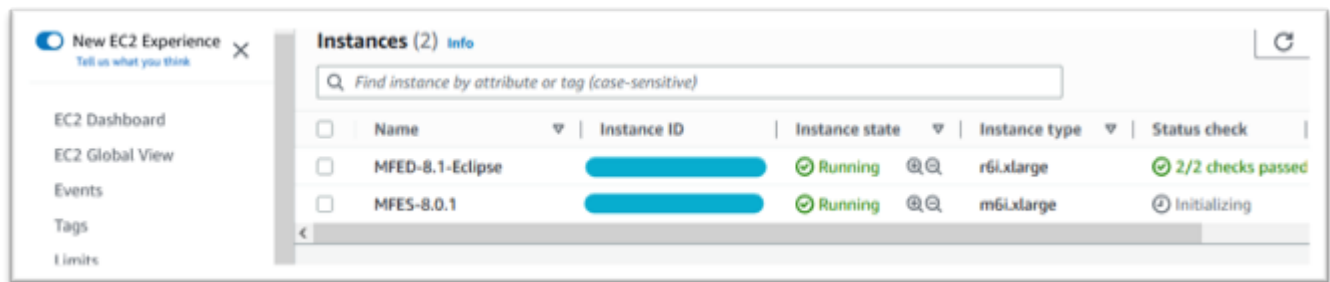
Wählen Sie in diesem Fall die Option Instanzkonfiguration bearbeiten und ändern Sie den Instanztyp.



14. Sobald die Erfolgsmeldung angezeigt wird, wählen Sie Mit Instanz verbinden, um Verbindungsdetails zu erhalten.



15. Navigieren Sie alternativ zu EC2 in der AWS Management Console.
16. Wählen Sie Instances, um den Status der neuen Instance zu sehen.



## Subnetz oder VPC ohne Internetzugang

Nehmen Sie diese zusätzlichen Änderungen vor, wenn das Subnetz oder die VPC keinen ausgehenden Internetzugang hat.

Der Lizenzmanager benötigt Zugriff auf die folgenden AWS-Services:

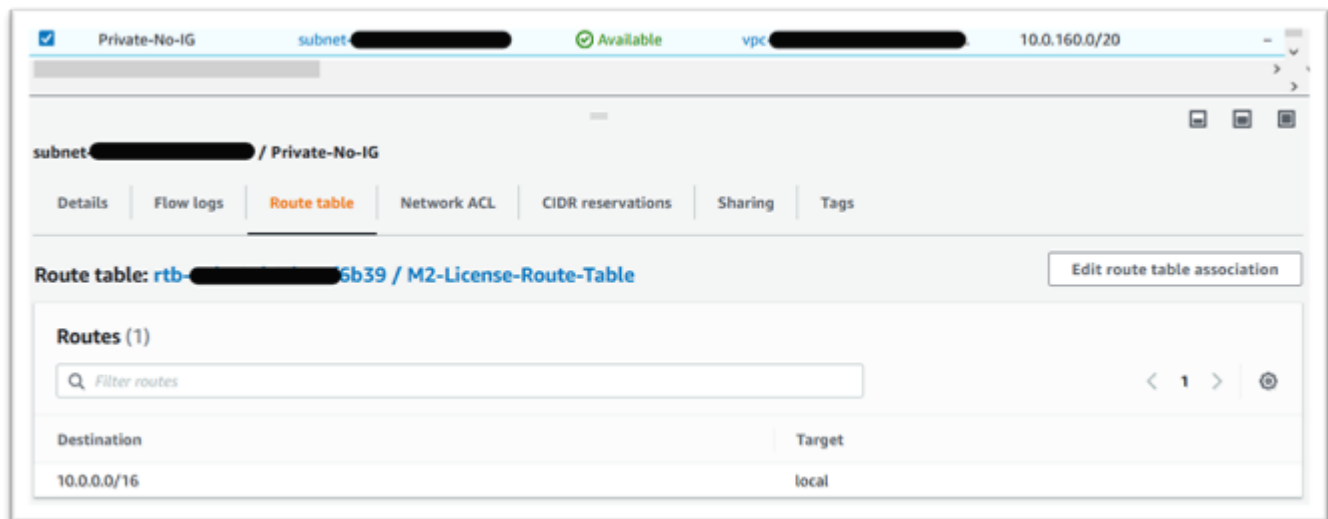
- `com.amazonaws.region.s3`
- `com.amazonaws.region.ec2`
- `com.amazonaws.region.license-manager`
- `com.amazonaws.region.sts`

In den vorherigen Schritten wurde die Datei `com.amazonaws` definiert. *Region* .s3-Dienst als Gateway-Endpunkt. Dieser Endpunkt benötigt einen Routing-Tabelleneintrag für alle Subnetze ohne Internetzugang.

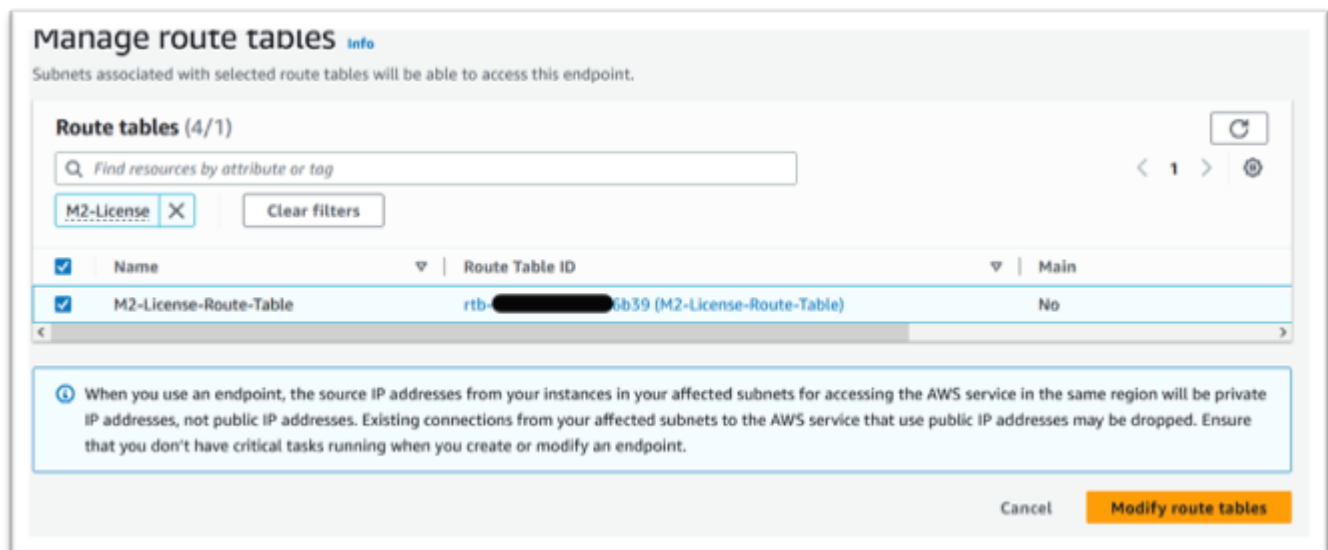
Die zusätzlichen drei Dienste werden als Schnittstellen-Endpunkte definiert.

Fügen Sie den Routen-Tabelleneintrag für den Amazon S3 S3-Endpunkt hinzu

1. Navigieren Sie zu VPC in AWS Management Console und wählen Sie Subnetze aus.
2. Wählen Sie das Subnetz aus, in dem die Amazon EC2 EC2-Instances erstellt werden sollen, und wählen Sie die Registerkarte Route Table.
3. Notieren Sie sich einige nachstehende Ziffern der Route-Tabellen-ID. Zum Beispiel die 6b39 in der Abbildung unten.



4. Wählen Sie im Navigationsbereich Endpoints aus.
5. Wählen Sie den zuvor erstellten Endpunkt und anschließend Routentabellen verwalten aus, entweder auf der Registerkarte Routentabellen für den Endpunkt oder in der Dropdownliste Aktionen.
6. Wählen Sie die Routentabelle anhand der zuvor identifizierten Ziffern aus und klicken Sie auf Routentabellen ändern.



## Definieren Sie die erforderliche Sicherheitsgruppe

Die Amazon EC2- AWS STS, und License Manager Manager-Dienste kommunizieren über HTTPS über Port 443. Diese Kommunikation ist bidirektional und erfordert Regeln für eingehenden und ausgehenden Datenverkehr, damit die Instance mit den Services kommunizieren kann.

1. Navigieren Sie zu Amazon VPC in der AWS Management Console.
2. Suchen Sie in der Navigationsleiste nach Sicherheitsgruppen und wählen Sie Sicherheitsgruppe erstellen aus.
3. Geben Sie einen Namen und eine Beschreibung der Sicherheitsgruppe ein, zum Beispiel „Eingehend/Ausgehendes HTTPS“.
4. Drücken Sie das X im VPC-Auswahlbereich, um die Standard-VPC zu entfernen, und wählen Sie die VPC aus, die den S3-Endpunkt enthält.
5. Fügen Sie eine Regel für eingehenden Datenverkehr hinzu, die TCP-Verkehr auf Port 443 von überall aus zulässt.

### Note

Die Regeln für eingehenden (und ausgehenden Datenverkehr) können weiter eingeschränkt werden, indem die Quelle eingeschränkt wird. Weitere Informationen finden Sie unter [Steuern des Datenverkehrs zu Ihren AWS Ressourcen mithilfe von Sicherheitsgruppen](#) im Amazon VPC-Benutzerhandbuch.

The screenshot displays the AWS Management Console interface for configuring a security group rule. It is divided into two main sections: 'Basic details' and 'Inbound rules'.

**Basic details:**

- Security group name:** Inbound-Outbound HTTPS (with an 'info' link and a note: 'Name cannot be edited after creation.')
- Description:** Allow HTTPS traffic on port 443 (with an 'info' link)
- VPC:** A dropdown menu showing a selected VPC (with an 'info' link and a search icon)

**Inbound rules:**

Type	Protocol	Port range	Source	Description - optional
Custom TCP	TCP	443	Anywh...	HTTPS traffic

Below the table, there is an 'Add rule' button and a search box containing '0.0.0.0/0' with a close button.

6. Drücken Sie auf Sicherheitsgruppe erstellen.

## Erstellen Sie die Dienstendpunkte

Wiederholen Sie diesen Vorgang dreimal — einmal für jeden Dienst.



1. Navigieren Sie im zu Amazon VPC AWS Management Console und wählen Sie Endpoints aus.
2. Drücken Sie auf Endpunkt erstellen.
3. Geben Sie einen Namen ein, zum Beispiel „Micro-Focus-License-EC2“, „Micro-Focus-License-STS“ oder „Micro-Focus-License-Manager“.
4. Wählen Sie die AWS-Servicekategorie aus.

**Endpoint settings**

**Name tag - optional**  
Creates a tag with a key of 'Name' and a value that you specify.

Micro-Focus-License-EC2

**Service category**  
Select the service category

**AWS services**  
Services provided by Amazon

**PrivateLink Ready partner services**  
Services with an AWS Service Ready designation

**AWS Marketplace services**  
Services that you've purchased through AWS Marketplace

**Other endpoint services**  
Find services shared with you by service name

5. Suchen Sie unter Dienste nach dem passenden Interface-Dienst, der einer der folgenden ist:
  - „com.amazonaws. *region* .ec2“
  - „com.amazonaws. *region* .sts“
  - „com.amazonaws. *region* .license-manager“

Beispielsweise:

- „com.amazonaws.us-west-1.ec2“
  - „com.amazonaws.us-west-1.sts“
  - „com.amazonaws.us-west-1.lizenzmanager“
6. Wählen Sie den passenden Interface-Dienst aus.

com.amazonaws. *region* .ec2:

The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'com.amazonaws.us-west-1.ec2'. The table below shows two services:

Service Name	Owner	Type
com.amazonaws.us-west-1.ec2	amazon	Interface
com.amazonaws.us-west-1.ec2messages	amazon	Interface

com.amazonaws. **region** .sts:

The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'Service Name = com.amazonaws.us-west-1.sts'. The table below shows one service:

Service Name	Owner	Type
com.amazonaws.us-west-1.sts	amazon	Interface

com.amazonaws. **region** .license-manager:


The screenshot shows the AWS IAM console 'Services' page. The search bar contains 'Find resources by attribute or tag'. A filter is applied: 'Service Name = com.amazonaws.us-west-1.license-manager'. The table below shows one service:

Service Name	Owner	Type
com.amazonaws.us-west-1.license-manager	amazon	Interface

7. Wählen Sie für VPC die VPC für die Instance aus.

**VPC**  
Select the VPC in which to create the endpoint

**VPC**  
The VPC in which to create your endpoint.

vpc-██████████ (Modernization-vpc1) ▼ 

▶ Additional settings

8. Wählen Sie die Availability Zone und die Subnetze für die VPC aus.

**Subnets ( 1/2 )** [Info](#)

<input type="checkbox"/>	Availability Zone ▼	Subnet ID ▼
<input checked="" type="checkbox"/>	us-west-1b (usw1-az3)	subnet-██████████ ▼
<input type="checkbox"/>	us-west-1c (usw1-az1)	██████████ ▼

subnet-██████████ ✕  
Private-No-IG

IP address type

IPv4


IPv6

Dualstack

9. Wählen Sie die zuvor erstellte Sicherheitsgruppe aus.

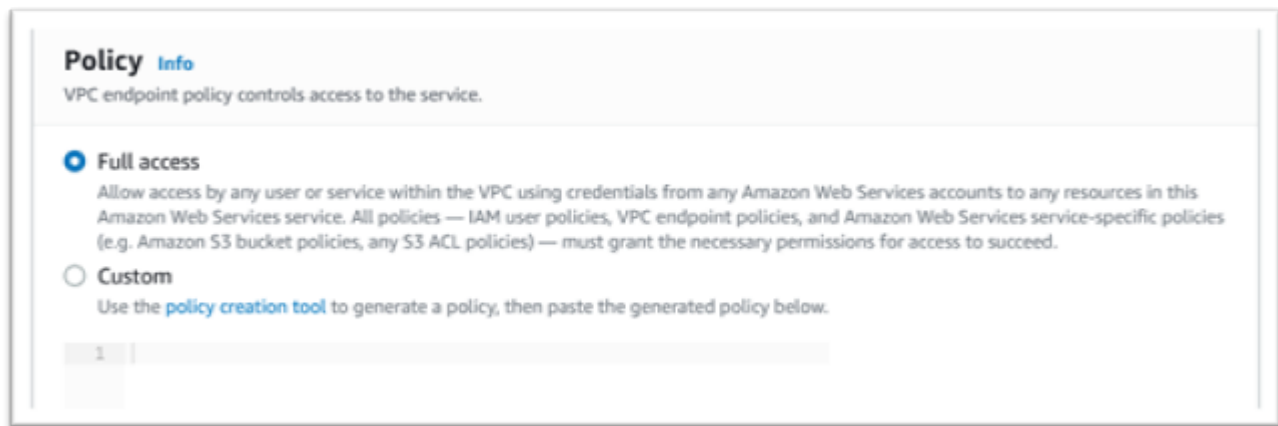
**Security groups ( 1 )** [Info](#)

🔍 Find resources by attribute or tag

Group name = Inbound-Outbound HTTPS ✕ 

<input type="checkbox"/>	Group ID ▼	Group name ▼
<input type="checkbox"/>	sg-██████████	Inbound-Outbound HTTPS

10. Wählen Sie unter Richtlinie die Option Vollzugriff aus.



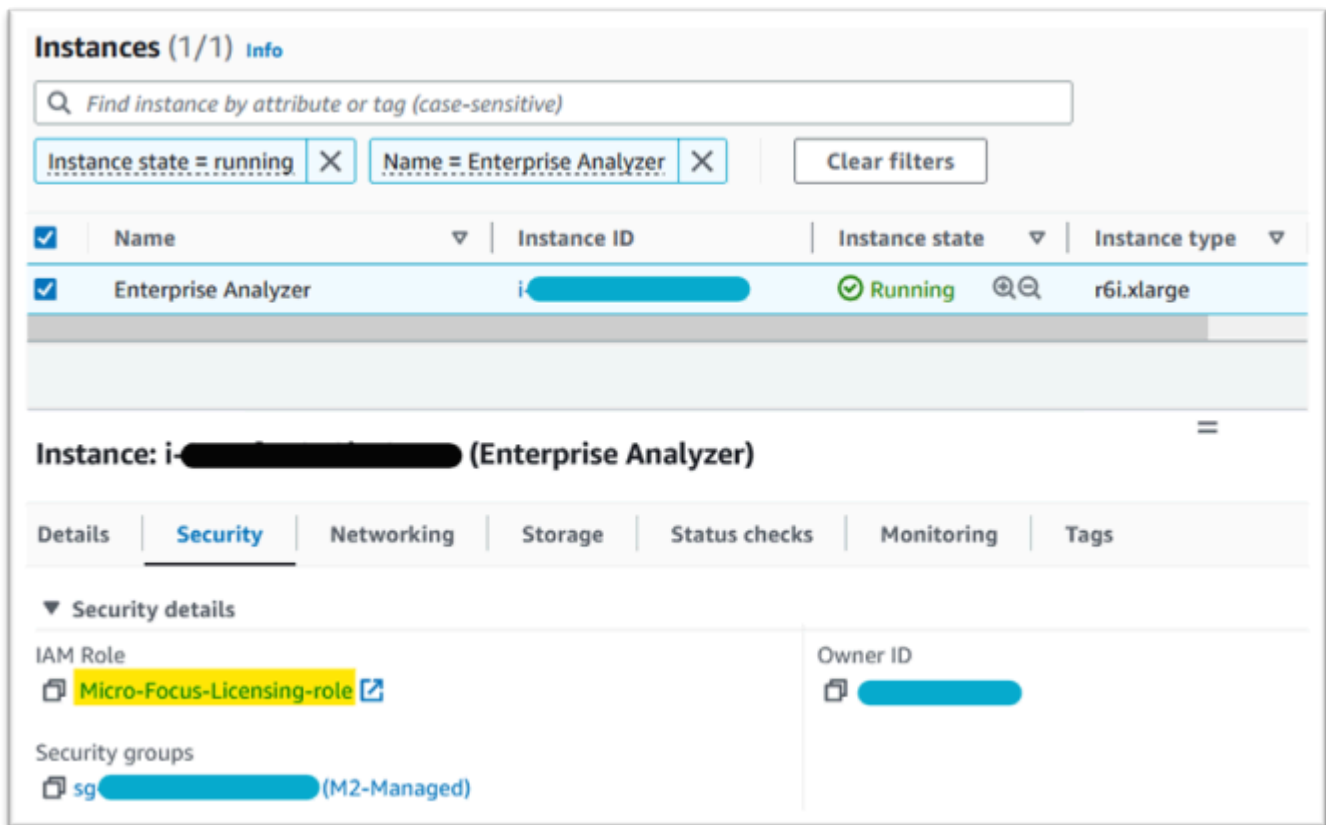
11. Klicken Sie auf Endpunkt erstellen.
12. Wiederholen Sie diesen Vorgang für die übrigen Schnittstellen.

## Behebung von Lizenzproblemen

Wenn Sie Probleme beim Zugriff auf oder bei der Verwendung der AMIs haben, können Ihnen die folgenden Informationen weiterhelfen.

Stellen Sie sicher, dass die Amazon EC2 EC2-Instance die IAM-Lizenzierungsrolle hat

Dies kann auf der Registerkarte Sicherheit der Amazon EC2 EC2-Instance-Details überprüft werden. Dies kann mithilfe der Sicherheitsoption im Drop-down-Menü Aktionen geändert werden.



## Verwenden Sie den Reachability Analyzer

Suchen Sie den Reachability Analyzer auf der AWS Network Manager Konsolenseite.

Erstellen und analysieren Sie einen Pfad zwischen der Amazon EC2 EC2-Instance, die über das AMI erstellt wurde, und dem Amazon S3 S3-VPC-Endpunkt.

Wenn die Amazon EC2 EC2-Instance keinen Internetzugang hat, wiederholen Sie die Pfadanalyse zu allen 4 Endpunkten.

Weitere Informationen zum Reachability Analyzer finden Sie unter [Erste Schritte mit Reachability Analyzer im Reachability Analyzer-Handbuch](#).

## Führen Sie den License-Daemon aus

Verwenden Sie unter Windows Enterprise Developer den folgenden Befehl von einer Befehlszeile aus:

```
"C:\Program Files (x86)\Micro Focus\Enterprise Developer\AdoptOpenJDK\bin\java" -jar
"C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

und untersuchen Sie die Ausgabe. Ignorieren Sie die SLF4J-Meldungen und suchen Sie nach der ersten Ausnahme.

Verwenden Sie auf Enterprise Analyzer den folgenden Befehl von einer Befehlszeile aus:

```
"C:\Program Files (x86)\Micro Focus\AdoptOpenJDK\bin\java" -jar "C:\Program Files (x86)\Micro Focus\Licensing\aws-license-daemon.jar"
```

und untersuchen Sie die Ausgabe. Ignorieren Sie die SLF4J-Meldungen und suchen Sie nach der ersten Ausnahme.

Unter Linux starte:

```
java -jar /var/microfocuslicensing/bin/aws-license-daemon.jar
```

Ignoriere die SLF4J-Meldungen und suche nach der ersten Ausnahme.

Wenn die Amazon S3 S3-Ressource beispielsweise nicht verfügbar ist, lautet die Ausnahme wie folgt:

```
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
```

Die Ausnahmemeldung gibt an, welche Ressource nicht verfügbar ist. Vergleichen Sie die Konfigurationswerte mit den Werten in diesem Thema.

## Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten

AWS Mainframe Modernization bietet über Amazon AppStream 2.0 mehrere Tools. AppStream 2.0 ist ein vollständig verwalteter, sicherer Anwendungs-Streaming-Dienst, mit dem Sie Desktop-Anwendungen an Benutzer streamen können, ohne Anwendungen neu schreiben zu müssen. AppStream 2.0 bietet Benutzern sofortigen Zugriff auf die Anwendungen, die sie benötigen, und bietet

eine reaktionsschnelle, flüssige Benutzererfahrung auf dem Gerät ihrer Wahl. Die Verwendung von AppStream 2.0 zum Hosten von Runtime Engine-spezifischen Tools gibt den Anwendungsteams von Kunden die Möglichkeit, die Tools direkt von ihren Webbrowsern aus zu verwenden und mit Anwendungsdateien zu interagieren, die entweder in Amazon S3 S3-Buckets oder Repositories gespeichert sind. CodeCommit

Informationen zur Browserunterstützung in AppStream 2.0 finden Sie unter [Systemanforderungen und Funktionsunterstützung \(Webbrowser\)](#) im Amazon AppStream 2.0-Administrationshandbuch. Wenn Sie bei der Verwendung von AppStream 2.0 Probleme haben, finden Sie weitere Informationen unter [Troubleshooting AppStream 2.0-Benutzerprobleme](#) im Amazon AppStream 2.0-Administrationshandbuch.

Dieses Dokument richtet sich an Mitglieder des Kundenserviceteams. Es wird beschrieben, wie Amazon AppStream 2.0-Flotten und -Stacks eingerichtet werden, um die im Rahmen der AWS Mainframe-Modernisierung verwendeten Tools Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer zu hosten. Micro Focus Enterprise Analyzer wird in der Regel während der Bewertungsphase und Micro Focus Enterprise Developer in der Regel während der Migrations- und Modernisierungsphase des AWS Mainframe-Modernisierungsansatzes verwendet. Wenn Sie sowohl Enterprise Analyzer als auch Enterprise Developer verwenden möchten, müssen Sie für jedes Tool separate Flotten und Stacks erstellen. Jedes Tool benötigt eine eigene Flotte und einen eigenen Stack, da die Lizenzbedingungen unterschiedlich sind.

#### Important

Die Schritte in diesem Tutorial basieren auf der herunterladbaren AWS CloudFormation Vorlage [cfn-m2-appstream-fleet-ea-ed.yml](#).

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Holen Sie sich die AppStream 2.0-Bilder](#)
- [Schritt 2: Erstellen Sie den Stack mithilfe der Vorlage AWS CloudFormation](#)
- [Schritt 3: Erstellen Sie einen Benutzer in AppStream 2.0](#)
- [Schritt 4: Melden Sie sich bei AppStream 2.0 an](#)
- [Schritt 5: Buckets in Amazon S3 verifizieren \(optional\)](#)
- [Nächste Schritte](#)

- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

- Laden Sie die Vorlage herunter: [cfn-m2-appstream-fleet-ea-ed.yml](#).
- Rufen Sie die ID Ihrer Standard-VPC und Sicherheitsgruppe ab. Weitere Informationen zur Standard-VPC finden Sie unter [Standard-VPCs](#) im Amazon VPC-Benutzerhandbuch. Weitere Informationen zur Standardsicherheitsgruppe finden Sie unter [Standard- und benutzerdefinierte Sicherheitsgruppen](#) im Amazon EC2 EC2-Benutzerhandbuch.
- Stellen Sie sicher, dass Sie über die folgenden Berechtigungen verfügen:
  - Erstellen Sie Stacks, Flotten und Benutzer in AppStream 2.0.
  - Stapel AWS CloudFormation mithilfe einer Vorlage erstellen.
  - Erstellen Sie Buckets und laden Sie Dateien in Buckets in Amazon S3 hoch.
  - Anmeldeinformationen (access\_key\_id und secret\_access\_key) von IAM herunterladen.

## Schritt 1: Holen Sie sich die AppStream 2.0-Bilder

In diesem Schritt teilen Sie die AppStream 2.0-Images für Enterprise Analyzer und Enterprise Developer mit Ihrem AWS Konto.


1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der linken Navigationsleiste Tools aus.
3. Wählen Sie unter Assets analysieren, entwickeln und erstellen die Option Assets mit meinem AWS Konto teilen aus.

## Schritt 2: Erstellen Sie den Stack mithilfe der Vorlage AWS CloudFormation

In diesem Schritt verwenden Sie die heruntergeladene AWS CloudFormation Vorlage, um einen AppStream 2.0-Stack und eine Flotte für die Ausführung von Micro Focus Enterprise Analyzer zu erstellen. Sie können diesen Schritt später wiederholen, um einen weiteren AppStream 2.0-Stack und eine weitere Flotte für die Ausführung von Micro Focus Enterprise Developer zu erstellen, da jedes Tool eine eigene Flotte und einen eigenen Stack in AppStream 2.0 benötigt. Weitere Informationen



zu AWS CloudFormation Stacks finden Sie unter [Arbeiten mit Stacks](#) im AWS CloudFormation Benutzerhandbuch.

 Note

AWS Bei der Mainframe-Modernisierung fällt für die Nutzung von Enterprise Analyzer und Enterprise Developer eine zusätzliche Gebühr zum Standard AppStream 2.0-Preis an. Weitere Informationen finden Sie unter Preise für die [AWS Mainframe-Modernisierung](#).

1. Laden Sie bei Bedarf die Vorlage [cfn-m2-appstream-fleet-ea-ed.yml](#) herunter.
2. Öffnen Sie die Konsole und wählen Sie Create Stack and with new resources (Standard). AWS CloudFormation
3. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus.
4. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen aus.
5. Wählen Sie unter Vorlagendatei hochladen die Option Datei auswählen aus und laden Sie die Vorlage [cfn-m2-appstream-fleet-ea-ed.yml](#) hoch.
6. Wählen Sie Weiter aus.

CloudFormation > Stacks > Create stack

Step 1  
**Specify template**

Step 2  
Specify stack details

Step 3  
Configure stack options

Step 4  
Review

## Create stack

**Prerequisite - Prepare template**

**Prepare template**  
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready  Use a sample template  Create template in Designer

**Specify template**  
A template is a JSON or YAML file that describes your stack's resources and properties.

**Template source**  
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL  Upload a template file

**Upload a template file**

`cfn-m2-appstream-fleet-ea-ed.yaml`

JSON or YAML formatted file

S3 URL: `https://s3-us-west-2.amazonaws.com/cf-templates-urr2587ffqs0-us-west-2/2022084KOV-cfn-m2-appstream-fleet-ea-ed.yaml`

7. Geben Sie unter Stack-Details an die folgenden Informationen ein:

- Geben Sie im Feld Stackname einen Namen Ihrer Wahl ein. z. B. **m2-ea**.
- Wählen Sie AppStream unter Anwendung die Option ea aus.
- Wählen Sie unter AppStreamFleetSecurityGruppe die Standardsicherheitsgruppe Ihrer Standard-VPC aus.
- Wählen Sie unter AppStreamFleetVpcSubnetz ein Subnetz innerhalb Ihrer Standard-VPC aus.
- Wählen Sie unter AppStreamImageNamedas Bild aus, das mit beginnt. `m2-enterprise-analyzer` Dieses Bild enthält die derzeit unterstützte Version des Micro Focus Enterprise Analyzer-Tools.
- Akzeptieren Sie die Standardwerte für die anderen Felder und wählen Sie dann Weiter.

Step 1  
**Specify template**


Step 2  
**Specify stack details**

Step 3  
Configure stack options

Step 4  
Review


## Specify stack details


**Stack name**

Stack name 


Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).


**Parameters**  
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

**AppStreamApplication**   
AppStream application

**AppStreamFleetSecurityGroup**   
AppStream fleet security group

**AppStreamFleetType**  
AppStream fleet type

**AppStreamFleetVpcSubnet**   
AppStream fleet subnet

**AppStreamImageName**   
AppStream machine image name: m2-enterprise-analyzer-v7.0.1.R1 or m2-enterprise-developer-v7.0.3.R1

**AppStreamInstanceType**  
AppStream instance type

**AppStreamInstances**  
AppStream desired instances

**AppStreamView**  
AppStream view

Cancel Previous **Next**

- Akzeptieren Sie alle Standardwerte und wählen Sie dann erneut Weiter.
- Stellen Sie bei der Überprüfung sicher, dass alle Parameter Ihren Vorstellungen entsprechen.
- Scrollen Sie nach unten, wählen Sie Ich bestätige, dass AWS IAM-Ressourcen mit benutzerdefinierten Namen erstellen CloudFormation könnte, und wählen Sie Create Stack aus.

Es dauert zwischen 20 und 30 Minuten, bis der Stack und die Flotte erstellt sind. Sie können „Aktualisieren“ wählen, um die AWS CloudFormation Ereignisse zu sehen, sobald sie eintreten.

## Schritt 3: Erstellen Sie einen Benutzer in AppStream 2.0

Während Sie darauf warten, AWS CloudFormation dass die Erstellung des Stacks abgeschlossen ist, können Sie in AppStream 2.0 einen oder mehrere Benutzer erstellen. Diese Benutzer sind diejenigen, die Enterprise Analyzer in AppStream 2.0 verwenden werden. Sie müssen für jeden Benutzer eine E-Mail-Adresse angeben und sicherstellen, dass jeder Benutzer über ausreichende Berechtigungen verfügt, um Buckets in Amazon S3 zu erstellen, Dateien in einen Bucket hochzuladen und einen Link zu einem Bucket zu erstellen, um dessen Inhalt zuzuordnen.

1. Öffnen Sie die AppStream 2.0-Konsole.
2. Wählen Sie in der linken Navigationsleiste Benutzerpool aus.
3. Wählen Sie Create user (Benutzer erstellen) aus.
4. Geben Sie eine E-Mail-Adresse, an die der Benutzer eine E-Mail-Einladung zur Nutzung von AppStream 2.0 erhalten kann, sowie einen Vor- und Nachnamen an und wählen Sie Benutzer erstellen aus.
5. Wiederholen Sie den Vorgang bei Bedarf, um weitere Benutzer zu erstellen. Die E-Mail-Adresse für jeden Benutzer muss eindeutig sein.

Weitere Informationen zum Erstellen von AppStream 2.0-Benutzern finden Sie unter [AppStream 2.0-Benutzerpools](#) im Amazon AppStream 2.0-Administrationshandbuch.

Wenn Sie mit der Erstellung des Stacks AWS CloudFormation fertig sind, können Sie den Benutzer, den Sie erstellt haben, dem Stack wie folgt zuweisen:

1. Öffnen Sie die AppStream 2.0-Konsole.
2. Wählen Sie den Benutzernamen.
3. Wählen Sie Aktion und dann Stapel zuweisen.
4. Wählen Sie unter Stapel zuweisen den Stapel aus, der mit `beginntm2-appstream-stack-ea`.
5. Wählen Sie Assign stack.

**Assign stack** ✕

Select a stack to enable access to the user(s) below.

User(s) being assigned

- Mary Major (mary.major@example.com)

Stack

m2-appstream-stack-ea-c92d75b0 ▼

Send email notification to user

Cancel **Assign stack**

Wenn Sie einem Stack einen Benutzer zuweisen, sendet AppStream 2.0 eine E-Mail an den Benutzer an die von Ihnen angegebene Adresse. Diese E-Mail enthält einen Link zur AppStream 2.0-Anmeldeseite.

## Schritt 4: Melden Sie sich bei AppStream 2.0 an

In diesem Schritt melden Sie sich bei AppStream 2.0 an, indem Sie den Link in der E-Mail verwenden, die von AppStream 2.0 an den Benutzer gesendet wurde, den Sie erstellt haben [Schritt 3: Erstellen Sie einen Benutzer in AppStream 2.0](#).

1. Melden Sie sich über den Link in der von AppStream 2.0 gesendeten E-Mail bei AppStream 2.0 an.
2. Ändern Sie Ihr Passwort, wenn Sie dazu aufgefordert werden. Der AppStream 2.0-Bildschirm, den Sie sehen, ähnelt dem folgenden:



3. Wählen Sie Desktop.
4. Wählen Sie in der Taskleiste Suchen und geben Sie die Eingabetaste ein **D** , um zum Home-Ordner zu navigieren.

**Note**

Wenn Sie diesen Schritt überspringen, wird möglicherweise eine Device not ready Fehlermeldung angezeigt, wenn Sie versuchen, auf den Basisordner zuzugreifen.

Wenn Sie Probleme bei der Anmeldung bei AppStream 2.0 haben, können Sie Ihre AppStream 2.0-Flotte jederzeit neu starten und versuchen, sich mithilfe der folgenden Schritte erneut anzumelden.

1. Öffnen Sie die AppStream 2.0-Konsole.
2. Wählen Sie in der linken Navigationsleiste Flotten aus.
3. Wählen Sie die Flotte aus, die Sie verwenden möchten.
4. Wählen Sie „Aktion“ und anschließend „Stopp“.
5. Warten Sie, bis die Flotte stoppt.
6. Wählen Sie „Aktion“ und anschließend „Start“.

Dieser Vorgang kann etwa 10 Minuten dauern.

## Schritt 5: Buckets in Amazon S3 verifizieren (optional)

Eine der Aufgaben, die durch die AWS CloudFormation Vorlage, die Sie zur Erstellung des Stacks verwendet haben, erledigt wurde, bestand darin, zwei Buckets in Amazon S3 zu erstellen, die zum Speichern und Wiederherstellen von Benutzerdaten und Anwendungseinstellungen über Arbeitssitzungen hinweg erforderlich sind. Diese Buckets lauten wie folgt:

- Der Name beginnt mit `appstream2-`. Dieser Bucket ordnet Daten Ihrem Home-Ordner in AppStream 2.0 zu (`D:\PhotonUser\My Files\Home Folder`).

### Note

Der Home-Ordner ist für eine bestimmte E-Mail-Adresse eindeutig und wird von allen Flotten und Stacks eines bestimmten AWS Kontos gemeinsam genutzt. Der Name des Home-Ordners ist ein SHA256-Hash der E-Mail-Adresse des Benutzers und wird in einem Pfad gespeichert, der auf diesem Hash basiert.

- Der Name beginnt mit `appstream-app-settings-`. Dieser Bucket enthält Benutzersitzungsinformationen für AppStream 2.0 und umfasst Einstellungen wie Browser-Favoriten, IDE- und Anwendungsverbindungsprofile sowie Anpassungen der Benutzeroberfläche. Weitere Informationen finden Sie unter [So funktioniert die Persistenz von Anwendungseinstellungen](#) im Amazon AppStream 2.0-Administrationshandbuch.

Gehen Sie wie folgt vor, um zu überprüfen, ob die Buckets erstellt wurden:

1. Öffnen Sie die Amazon S3-Konsole.
2. Wählen Sie in der linken Navigationsleiste Buckets aus.
3. Geben Sie im Feld Buckets nach Namen suchen ein, um die **appstream** Liste zu filtern.

Wenn Sie die Buckets sehen, sind keine weiteren Maßnahmen erforderlich. Seien Sie sich nur bewusst, dass die Buckets existieren. Wenn Sie die Buckets nicht sehen, ist entweder die Ausführung der AWS CloudFormation Vorlage noch nicht abgeschlossen, oder es ist ein Fehler aufgetreten. Gehen Sie zur AWS CloudFormation Konsole und überprüfen Sie die Meldungen zur Stack-Erstellung.

## Nächste Schritte

Nachdem die AppStream 2.0-Infrastruktur eingerichtet ist, können Sie Enterprise Analyzer einrichten und mit der Nutzung beginnen. Weitere Informationen finden Sie unter [Tutorial: Enterprise Analyzer auf AppStream 2.0 einrichten](#). Sie können Enterprise Developer auch einrichten. Weitere Informationen finden Sie unter [Tutorial: Micro Focus Enterprise Developer auf AppStream 2.0 einrichten](#).

## Bereinigen von -Ressourcen

Das Verfahren zum Bereinigen des erstellten Stacks und der erstellten Flotten ist unter [Create an AppStream 2.0 Fleet and Stack](#) beschrieben.

Wenn die AppStream 2.0-Objekte gelöscht wurden, kann der Kontoadministrator gegebenenfalls auch die Amazon S3 S3-Buckets für Anwendungseinstellungen und Basisordner bereinigen.

### Note

Der Home-Ordner für einen bestimmten Benutzer ist für alle Flotten einzigartig. Daher müssen Sie ihn möglicherweise behalten, wenn andere AppStream 2.0-Stacks in demselben Konto aktiv sind.

Schließlich erlaubt es AppStream 2.0 derzeit nicht, Benutzer über die Konsole zu löschen. Stattdessen müssen Sie die Service-API mit der CLI verwenden. Weitere Informationen finden Sie unter [Benutzerpool-Administration](#) im Amazon AppStream 2.0-Administrationshandbuch.

## Tutorial: Enterprise Analyzer auf AppStream 2.0 einrichten

In diesem Tutorial wird beschrieben, wie Sie Micro Focus Enterprise Analyzer einrichten, um eine oder mehrere Mainframe-Anwendungen zu analysieren. Das Enterprise Analyzer-Tool bietet mehrere Berichte, die auf der Analyse des Anwendungsquellcodes und der Systemdefinitionen basieren.

Dieses Setup wurde entwickelt, um die Zusammenarbeit im Team zu fördern. Bei der Installation wird ein Amazon S3 S3-Bucket verwendet, um den Quellcode mit virtuellen Laufwerken zu teilen. Dabei wird [Rclone](#) () auf dem Windows-Computer verwendet. Mit einer gemeinsamen Amazon RDS-Instance, auf der [PostgreSQL](#) ausgeführt wird, kann jedes Mitglied des Teams auf alle angeforderten Berichte zugreifen.



Teammitglieder können die virtuelle Amazon S3 S3-gesicherte Festplatte auch auf ihren persönlichen Computern mounten und den Quell-Bucket von ihren Workstations aus aktualisieren. Sie können möglicherweise Skripts oder jede andere Form der Automatisierung auf ihren Computern verwenden, wenn sie mit anderen internen Systemen vor Ort verbunden sind.

Das Setup basiert auf den AppStream 2.0-Windows-Images, die AWS Mainframe Modernization dem Kunden zur Verfügung stellt. Die Einrichtung basiert außerdem auf der Erstellung von AppStream 2.0-Flotten und -Stacks, wie unter beschrieben. [Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten](#)

### Important

Bei den Schritten in diesem Tutorial wird davon ausgegangen, dass Sie AppStream 2.0 mit der herunterladbaren AWS CloudFormation Vorlage [cfn-m2-appstream-fleet-ea-ed.yml](#) einrichten. Weitere Informationen finden Sie unter [Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten](#).

Um die Schritte in diesem Tutorial ausführen zu können, müssen Sie Ihre Enterprise Analyzer-Flotte und Ihren Enterprise Analyzer-Stack eingerichtet haben und diese müssen laufen.

Eine vollständige Beschreibung der Funktionen und Leistungen von Enterprise Analyzer finden Sie in der [Enterprise Analyzer-Dokumentation](#) auf der Micro Focus-Website.

## Inhalt des Bilds

Neben der Enterprise Analyzer-Anwendung selbst enthält das Image die folgenden Tools und Bibliotheken.

### Tools von Drittanbietern

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [git-scm](#)
- [PostgreSQL ODBC-Treiber](#)

## Bibliotheken in C:\Users\Public

- BankDemo Quellcode und Projektdefinition für Enterprise Developer:m2-bankdemo-template.zip.
- MFA-Installationspaket für den Mainframe:. mfa.zip Weitere Informationen finden Sie unter [Mainframe Access Overview](#) in der Micro Focus Enterprise Developer-Dokumentation.
- Befehls- und Konfigurationsdateien für Rclone (Anweisungen zu deren Verwendung in den Tutorials): m2-rclone.cmd und. m2-rclone.conf

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Einrichtung](#)
- [Schritt 2: Erstellen Sie den Amazon S3 S3-basierten virtuellen Ordner unter Windows](#)
- [Schritt 3: Erstellen Sie eine ODBC-Quelle für die Amazon RDS-Instance](#)
- [Nachfolgende Sitzungen](#)
- [Problembehandlung bei der Workspace-Verbindung](#)
- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

- Laden Sie den Quellcode und die Systemdefinitionen für die Kundenanwendung, die Sie analysieren möchten, in einen S3-Bucket hoch. Zu den Systemdefinitionen gehören CICS CSD, DB2-Objektdefinitionen usw. Sie können innerhalb des Buckets eine Ordnerstruktur erstellen, die für die Organisation der Anwendungsartefakte sinnvoll ist. Wenn Sie das BankDemo Beispiel beispielsweise entpacken, hat es die folgende Struktur:

```
demo
  |--> jcl
  |--> RDEF
  |--> transaction
  |--> xa
```

- Erstellen und starten Sie eine Amazon RDS-Instance, auf der PostgreSQL ausgeführt wird. Diese Instance speichert die von Enterprise Analyzer erstellten Daten und Ergebnisse. Sie können diese Instanz mit allen Mitgliedern des Anwendungsteams teilen. Erstellen Sie außerdem ein leeres

Schema mit dem Namen `m2_ea` (oder einem anderen geeigneten Namen) in der Datenbank. Definieren Sie Anmeldeinformationen für autorisierte Benutzer, die es ihnen ermöglichen, Elemente in diesem Schema zu erstellen, einzufügen, zu aktualisieren und zu löschen. Sie können den Datenbanknamen, die URL des Serverendpunkts und den TCP-Port von der Amazon RDS-Konsole oder vom Kontoadministrator abrufen.

- Stellen Sie sicher, dass Sie den programmatischen Zugriff auf Ihre AWS-Konto eingerichtet haben. Weitere Informationen finden Sie unter [Programmatischer Zugriff](#) in der Allgemeinen Amazon Web Services-Referenz

## Schritt 1: Einrichtung

1. Starten Sie eine Sitzung mit AppStream 2.0 mit der URL, die Sie in der Willkommens-E-Mail von AppStream 2.0 erhalten haben.
2. Verwenden Sie Ihre E-Mail-Adresse als Benutzer-ID und definieren Sie Ihr permanentes Passwort.
3. Wählen Sie den Enterprise Analyzer-Stack aus.
4. Wählen Sie auf der AppStream 2.0-Menüseite Desktop aus, um den Windows-Desktop aufzurufen, auf dem die Flotte streamt.

## Schritt 2: Erstellen Sie den Amazon S3 S3-basierten virtuellen Ordner unter Windows

### Note

Wenn Sie Rclone bereits in der Vorschauversion der AWS Mainframe-Modernisierung verwendet haben, müssen Sie `m2-rclone.cmd` auf die neuere Version aktualisieren, die sich unter befindet. `C:\Users\Public`

1. Kopieren Sie die bereitgestellten `m2-rclone.cmd` Dateien `m2-rclone.conf` und mit `C:\Users\Public` dem `C:\Users\PhotonUser\My Files\Home` Folder Datei-Explorer in Ihren Home-Ordner.
2. Aktualisieren Sie die `m2-rclone.conf` Konfigurationsparameter mit Ihrem AWS Zugriffsschlüssel und dem entsprechenden Geheimnis sowie Ihrem AWS-Region.

```
[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. Nehmen Sie in `m2-rc1one.cmd` die folgenden Änderungen vor:

- Wechseln Sie `your-s3-bucket` zu Ihrem Amazon S3 S3-Bucket-Namen. z. B. `m2-s3-mybucket`.
- Wechseln Sie `your-s3-folder-key` zu Ihrem Amazon S3 S3-Bucket-Key. z. B. `myProject`.
- Wechseln Sie `your-local-folder-path` zum Pfad des Verzeichnisses, in das die Anwendungsdateien aus dem Amazon S3 S3-Bucket synchronisiert werden sollen, der sie enthält. z. B. `D:\PhotonUser\My Files\Home Folder\m2-new`. Dieses synchronisierte Verzeichnis muss ein Unterverzeichnis des Home-Ordners sein, damit AppStream 2.0 es beim Start und Ende der Sitzung ordnungsgemäß sichern und wiederherstellen kann.

```
:loop
timeout /T 10
"C:\Program Files\rclone\rclone.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rc1one.conf"
goto :loop
```

4. Öffnen Sie eine Windows-Eingabeaufforderung, geben Sie bei `C:\Users\PhotonUser\My Files\Home Folder` Bedarf eine `CD` ein und führen Sie den Befehl aus. `m2-rc1one.cmd` Dieses Befehlsskript führt eine Endlosschleife aus und synchronisiert Ihren Amazon S3 S3-Bucket und Ihren Schlüssel alle 10 Sekunden mit dem lokalen Ordner. Sie können das Timeout nach Bedarf anpassen. Sie sollten den Quellcode der Anwendung sehen, der sich im Amazon S3 S3-Bucket im Windows-Datei-Explorer befindet.

Um dem Set, an dem Sie gerade arbeiten, neue Dateien hinzuzufügen oder bestehende zu aktualisieren, laden Sie die Dateien in den Amazon S3 S3-Bucket hoch. Sie werden dann bei der

nächsten unter definierten Iteration mit Ihrem Verzeichnis synchronisiert. `m2-rc1one.cmd` Wenn Sie einige Dateien löschen möchten, löschen Sie sie ebenfalls aus dem Amazon S3 S3-Bucket. Beim nächsten Synchronisierungsvorgang werden sie aus Ihrem lokalen Verzeichnis gelöscht.

### Schritt 3: Erstellen Sie eine ODBC-Quelle für die Amazon RDS-Instance

1. Um das EA\_Admin-Tool zu starten, navigieren Sie zum Anwendungsauswahlmenü in der oberen linken Ecke des Browserfensters und wählen Sie MF EA\_Admin.
2. Wählen Sie im Menü „Verwalten“ die Option „ODBC-Datenquellen“ und anschließend auf der Registerkarte „Benutzer-DSN“ die Option „Hinzufügen“.
3. Wählen Sie im Dialogfeld Neue Datenquelle erstellen den PostgreSQL-Unicode-Treiber aus, und klicken Sie dann auf Fertig stellen.
4. Definieren Sie im Dialogfeld PostgreSQL Unicode ODBC Driver (psqlODBC) Setup den gewünschten Datenquellennamen und notieren Sie sich diesen. Vervollständigen Sie die folgenden Parameter mit den Werten aus der RDS-Instanz, die Sie zuvor erstellt haben:

#### Beschreibung

Optionale Beschreibung, die Ihnen hilft, diese Datenbankverbindung schnell zu identifizieren.

#### Datenbank

Die Amazon RDS-Datenbank, die Sie zuvor erstellt haben.

#### Server

Der Amazon RDS-Endpunkt.

#### Port

Der Amazon RDS-Port.

#### Benutzername

Wie in der Amazon RDS-Instance definiert.

#### Passwort

Wie in der Amazon RDS-Instance definiert.

5. Wählen Sie Test, um zu überprüfen, ob die Verbindung zu Amazon RDS erfolgreich ist, und wählen Sie dann Speichern, um Ihren neuen Benutzer-DSN zu speichern.

6. Warten Sie, bis Sie die Meldung sehen, die bestätigt, dass der richtige Workspace erstellt wurde, und wählen Sie dann OK, um mit ODBC-Datenquellen fertig zu werden und das EA\_Admin-Tool zu schließen.
7. Navigieren Sie erneut zum Anwendungsauswahlmenü und wählen Sie Enterprise Analyzer, um das Tool zu starten. Wählen Sie „Neu erstellen“.
8. Geben Sie im Fenster Workspace-Konfiguration Ihren Workspace-Namen ein und definieren Sie dessen Speicherort. Der Arbeitsbereich kann die Amazon S3 S3-basierte Festplatte sein, wenn Sie mit dieser Konfiguration arbeiten, oder Ihr Home-Ordner, wenn Sie dies bevorzugen.
9. Wählen Sie Andere Datenbank auswählen, um eine Verbindung zu Ihrer Amazon RDS-Instance herzustellen.
10. Wählen Sie das Postgre-Symbol aus den Optionen und klicken Sie dann auf OK.
11. Geben Sie für die Windows-Einstellungen unter Optionen — Verbindungsparameter definieren den Namen der Datenquelle ein, die Sie erstellt haben. Geben Sie außerdem den Datenbanknamen, den Schemanamen, den Benutzernamen und das Passwort ein. Wählen Sie OK aus.
12. Warten Sie, bis Enterprise Analyzer alle Tabellen, Indizes usw. erstellt hat, die zum Speichern der Ergebnisse benötigt werden. Dieser Vorgang kann einige Minuten dauern. Enterprise Analyzer bestätigt, wann die Datenbank und der Workspace einsatzbereit sind.
13. Navigieren Sie erneut zum Anwendungsauswahlmenü und wählen Sie Enterprise Analyzer, um das Tool zu starten.
14. Das Enterprise Analyzer-Startfenster wird am neuen, ausgewählten Workspace-Speicherort angezeigt. Wählen Sie OK aus.
15. Navigieren Sie im linken Bereich zu Ihrem Repository, wählen Sie den Repository-Namen und wählen Sie Dateien/Ordner zu Ihrem Workspace hinzufügen. Wählen Sie den Ordner aus, in dem Ihr Anwendungscode gespeichert ist, um ihn dem Workspace hinzuzufügen. Sie können den vorherigen BankDemo Beispielcode verwenden, wenn Sie möchten. Wenn Enterprise Analyzer Sie auffordert, diese Dateien zu verifizieren, wählen Sie Verify, um den ersten Enterprise Analyzer-Bestätigungsbericht zu starten. Je nach Größe Ihrer Anwendung kann es einige Minuten dauern, bis der Vorgang abgeschlossen ist.
16. Erweitere deinen Workspace, um die Dateien und Ordner zu sehen, die du dem Workspace hinzugefügt hast. Die Objekttypen und zyklomatischen Komplexitätsberichte sind auch im oberen Quadranten des Chart Viewer-Fensters sichtbar.

Sie können Enterprise Analyzer jetzt für alle erforderlichen Aufgaben verwenden.

## Nachfolgende Sitzungen

1. Starten Sie eine Sitzung mit AppStream 2.0 mit der URL, die Sie in der Willkommens-E-Mail von AppStream 2.0 erhalten haben.
2. Melden Sie sich mit Ihrer E-Mail-Adresse und Ihrem permanenten Passwort an.
3. Wählen Sie den Enterprise Analyzer-Stack aus.
4. Starten Sie `Rclone`, um eine Verbindung zur Amazon S3 S3-gesicherten Festplatte herzustellen, wenn Sie diese Option verwenden, um die Workspace-Dateien gemeinsam zu nutzen.
5. Starten Sie Enterprise Analyzer, um Ihre Aufgaben zu erledigen.

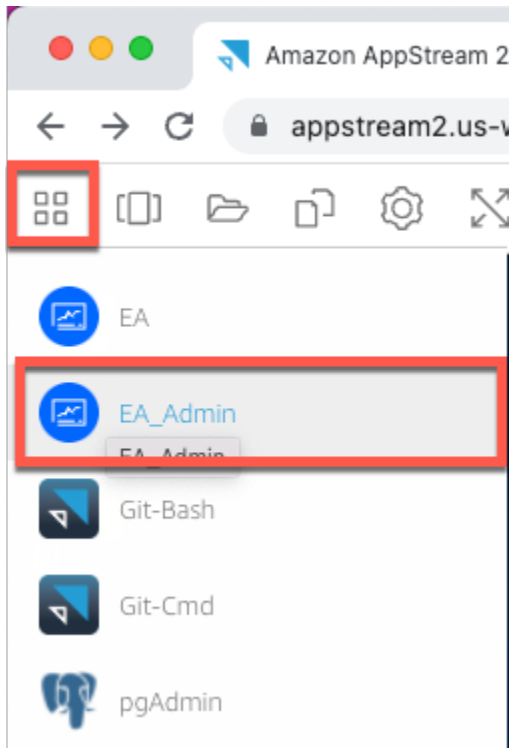
## Problembehandlung bei der Workspace-Verbindung

Wenn Sie versuchen, erneut eine Verbindung zu Ihrem Enterprise Analyzer-Workspace herzustellen, wird möglicherweise ein Fehler wie dieser angezeigt:

```
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?
```

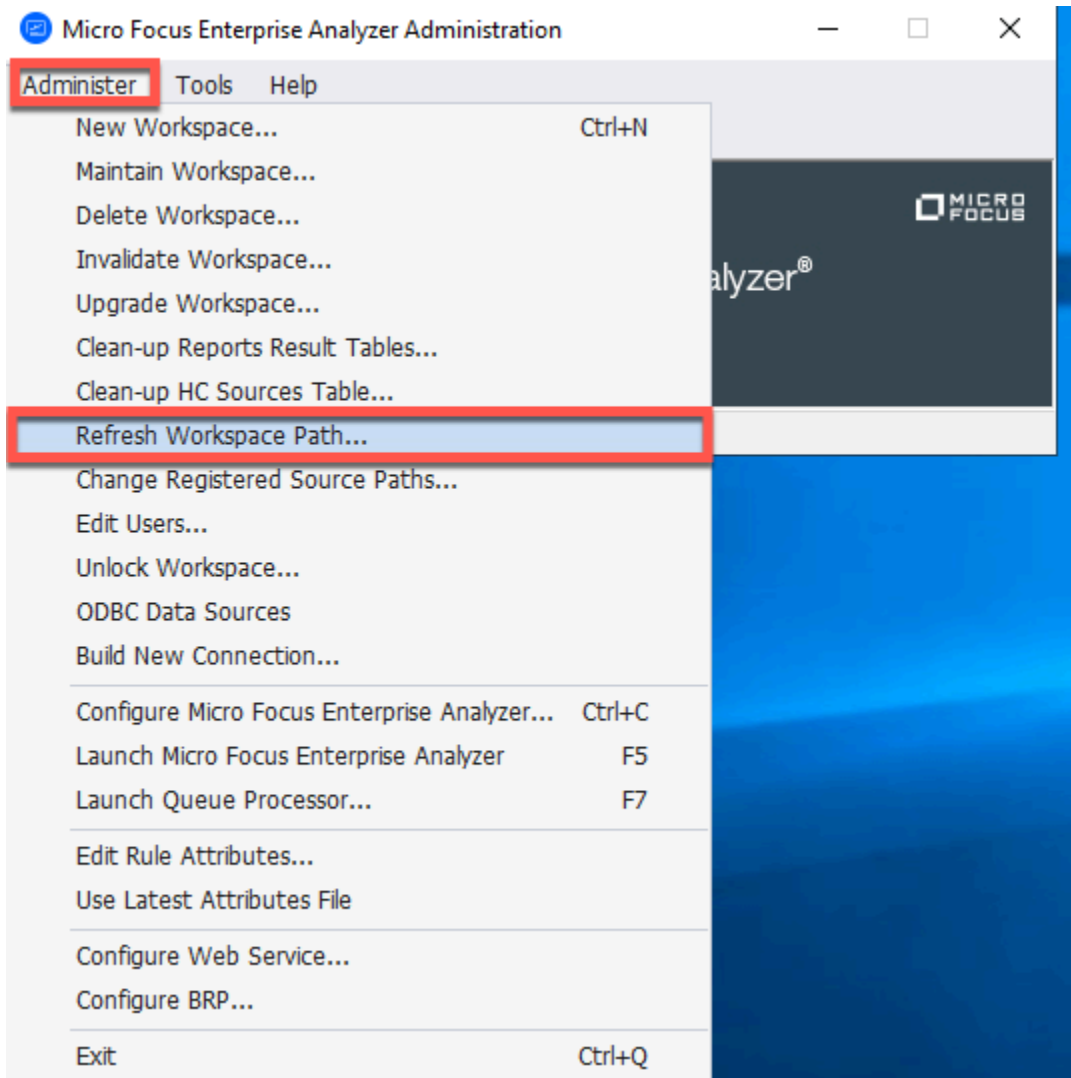
Um dieses Problem zu beheben, wählen Sie OK, um die Meldung zu löschen, und führen Sie dann die folgenden Schritte aus.

1. Wählen Sie in AppStream 2.0 in der Werkzeugleiste das Symbol „Anwendung starten“ und anschließend „EA\_Admin“, um das Micro Focus Enterprise Analyzer Administration Tool zu starten.

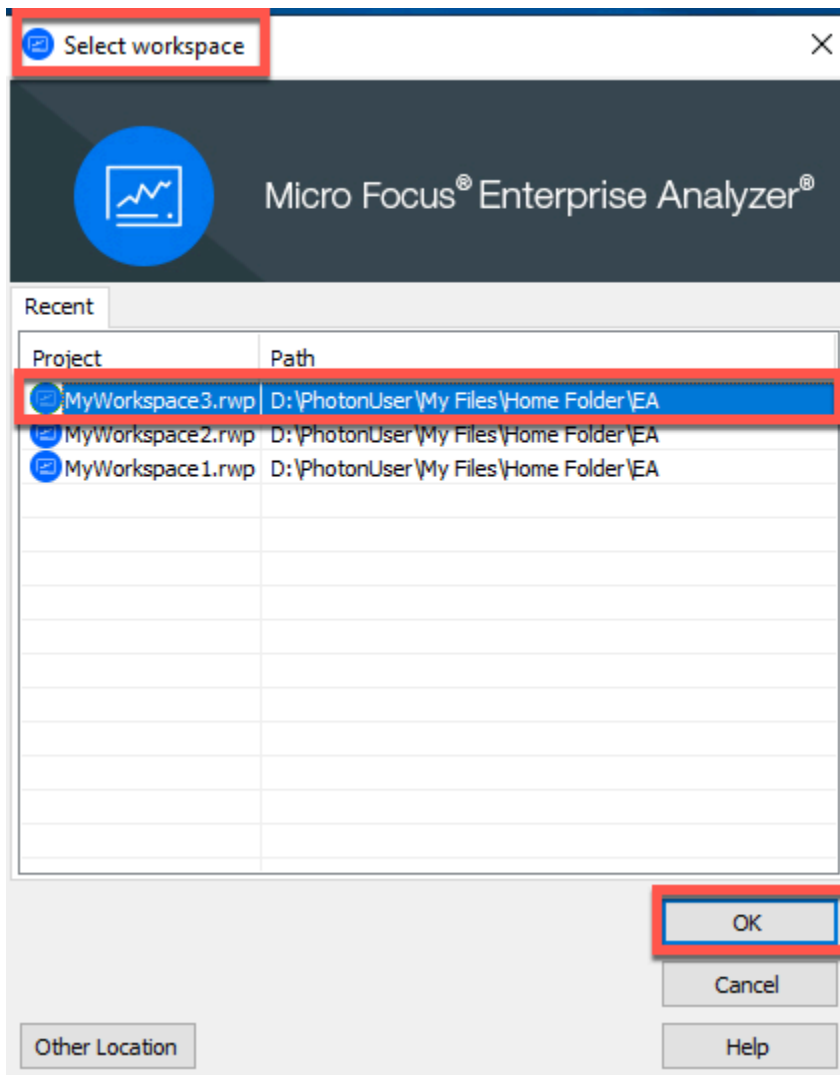


2. Wählen Sie im Menü „Verwalten“ die Option Workspace-Pfad aktualisieren... .

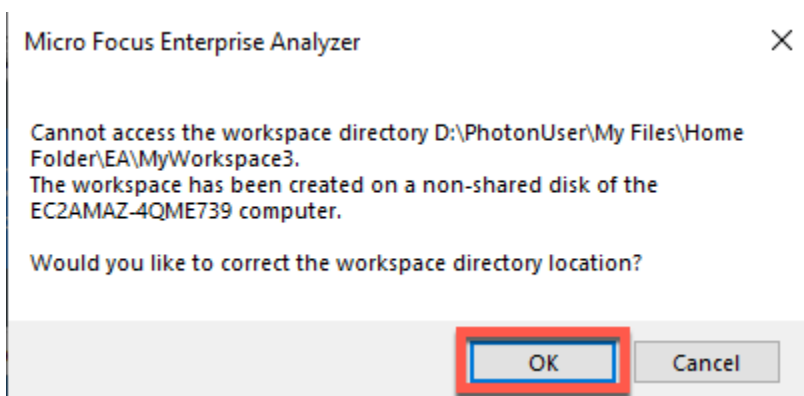




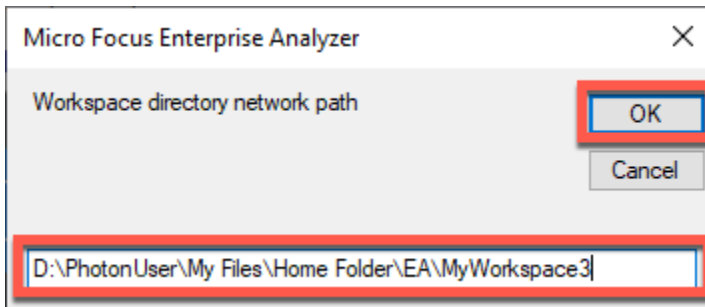
3. Wählen Sie unter Arbeitsbereich auswählen den gewünschten Arbeitsbereich aus, und klicken Sie dann auf OK.



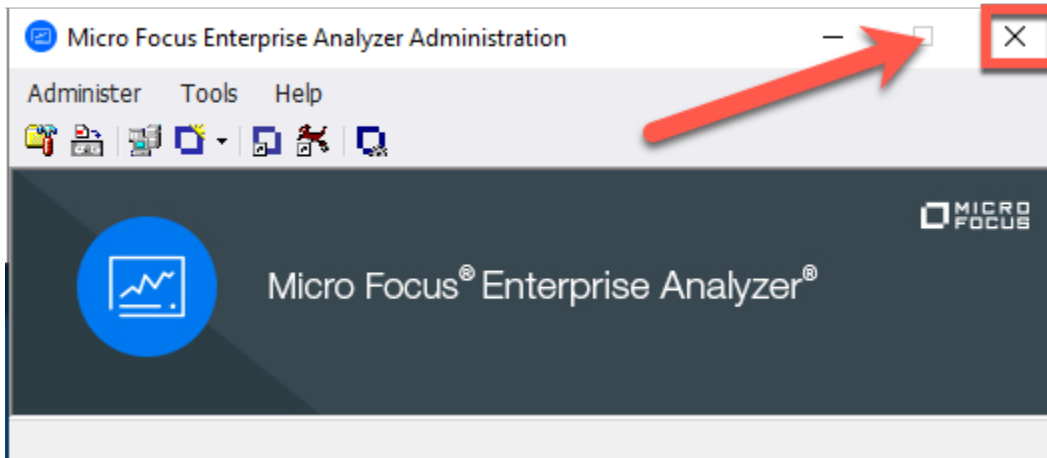
4. Wählen Sie OK, um die Fehlermeldung zu bestätigen.



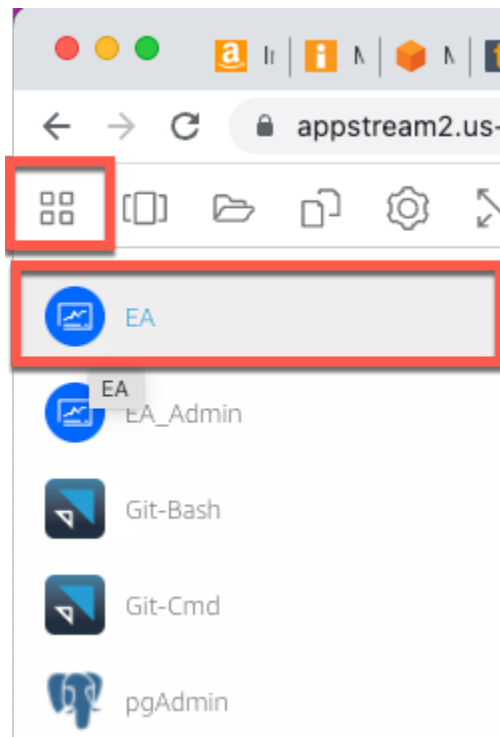
5. Geben Sie unter Netzwerkpfad für das Workspace-Verzeichnis den richtigen Pfad zu Ihrem Workspace ein, D:\PhotonUser\My Files\Home Folder\EA\MyWorkspace3 z. B.



- Schließen Sie das Micro Focus Enterprise Analyzer Administration Tool.



- Wählen Sie in AppStream 2.0 in der Werkzeugleiste das Symbol „Anwendung starten“ und anschließend „EA“, um Micro Focus Enterprise Analyzer zu starten.



## 8. Wiederhole die Schritte 3 bis 5.

Micro Focus Enterprise Analyzer sollte jetzt mit dem vorhandenen Workspace geöffnet werden.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, damit Ihnen keine weiteren Kosten entstehen. Führen Sie folgende Schritte aus:

- Verwenden Sie das EA\_Admin-Tool, um den Workspace zu löschen.
- Löschen Sie die S3-Buckets, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen eines Buckets](#) im Amazon S3 S3-Benutzerhandbuch.
- Löschen Sie die Datenbank, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen einer DB-Instance](#).

## Tutorial: Micro Focus Enterprise Developer auf AppStream 2.0 einrichten

In diesem Tutorial wird beschrieben, wie Sie Micro Focus Enterprise Developer für eine oder mehrere Mainframe-Anwendungen einrichten, um sie mithilfe der Enterprise Developer-Funktionen zu warten, zu kompilieren und zu testen. Das Setup basiert auf den AppStream 2.0-Windows-Images, die AWS Mainframe Modernization dem Kunden zur Verfügung stellt, und auf der Erstellung von AppStream 2.0-Flotten und -Stacks, wie unter beschrieben. [Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten](#)

### Important

Bei den Schritten in diesem Tutorial wird davon ausgegangen, dass Sie AppStream 2.0 mithilfe der herunterladbaren AWS CloudFormation Vorlage [cfn-m2-appstream-fleet-ea-ed.yaml](#) einrichten. Weitere Informationen finden Sie unter [Tutorial: AppStream 2.0 für die Verwendung mit Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer einrichten](#).

Sie müssen die Schritte dieses Setups durchführen, wenn die Enterprise Developer-Flotte und der Enterprise Developer-Stack betriebsbereit sind.

Eine vollständige Beschreibung der Funktionen und Leistungen von Enterprise Developer v7 finden Sie in der [up-to-date Online-Dokumentation \(v7.0\)](#) auf der Micro Focus-Website.

## Inhalt des Bilds

Zusätzlich zu Enterprise Developer selbst enthält das Image Rumba (einen TN3270-Emulator). Es enthält auch die folgenden Tools und Bibliotheken.

### Tools von Drittanbietern

- [Python](#)
- [Rclone](#)
- [pgAdmin](#)
- [git-scm](#)
- [PostgreSQL ODBC-Treiber](#)

### Bibliotheken in C:\Users\Public

- BankDemo Quellcode und Projektdefinition für Enterprise Developer:m2-bankdemo-template.zip.
- MFA-Installationspaket für den Mainframe:. mfa.zip Weitere Informationen finden Sie unter [Mainframe Access Overview](#) in der Micro Focus Enterprise Developer-Dokumentation.
- Befehls- und Konfigurationsdateien für Rclone (Anweisungen zu deren Verwendung in den Tutorials): m2-rclone.cmd und. m2-rclone.conf

Wenn Sie auf Quellcode zugreifen müssen, der noch nicht in CodeCommit Repositorys geladen wurde, der aber in einem Amazon S3 S3-Bucket verfügbar ist, um beispielsweise das erste Laden des Quellcodes in Git durchzuführen, gehen Sie wie unter beschrieben vor, um eine virtuelle Windows-Festplatte zu erstellen. [Tutorial: Enterprise Analyzer auf AppStream 2.0 einrichten](#)

### Themen

- [Voraussetzungen](#)
- [Schritt 1: Einrichtung durch einzelne Enterprise Developer-Benutzer](#)
- [Schritt 2: Erstellen Sie den Amazon S3-basierten virtuellen Ordner unter Windows \(optional\)](#)
- [Schritt 3: Klonen Sie das Repository](#)
- [Nachfolgende Sitzungen](#)

- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

- Ein oder mehrere CodeCommit Repositorys, die mit dem Quellcode der zu verwaltenden Anwendung geladen wurden. Das Repository-Setup sollte den Anforderungen der oben genannten CI/CD-Pipeline entsprechen, um durch die Kombination beider Tools Synergien zu schaffen.
- Jeder Benutzer muss über Anmeldeinformationen für das CodeCommit Repository oder die Repositorys verfügen, die vom Kontoadministrator gemäß den Informationen unter [Authentifizierung und Zugriffskontrolle für AWS CodeCommit](#) definiert wurden. Die Struktur dieser Anmeldeinformationen wird unter [Authentifizierung und Zugriffskontrolle für AWS](#) überprüft, CodeCommit und die vollständige Referenz zu den IAM-Autorisierungen für CodeCommit finden Sie in der [CodeCommit Berechtigungsreferenz](#): Der Administrator kann unterschiedliche IAM-Richtlinien für verschiedene Rollen definieren, deren Anmeldeinformationen für die jeweilige Rolle für jedes Repository spezifisch sind, und die Autorisierungen des Benutzers auf die spezifischen Aufgaben beschränken, die er in einem bestimmten Repository ausführen muss. Daher generiert der Kontoadministrator für jeden Betreuer des CodeCommit Repositorys einen Hauptbenutzer und erteilt diesem Benutzer Berechtigungen für den Zugriff auf das oder die erforderlichen Repositorys, indem er die richtige (n) IAM-Richtlinie (n) für den CodeCommit Zugriff auswählt.

## Schritt 1: Einrichtung durch einzelne Enterprise Developer-Benutzer

1. Besorgen Sie sich Ihre IAM-Anmeldeinformationen:
  1. Stellen Sie unter <https://console.aws.amazon.com/iam/> eine Connect zur AWS Konsole her.
  2. Folgen Sie dem Verfahren, das in Schritt 3 von [Setup für HTTPS-Benutzer mithilfe von Git-Anmeldeinformationen](#) im AWS CodeCommit Benutzerhandbuch beschrieben wird.
  3. Kopieren Sie die CodeCommit spezifischen Anmeldeinformationen, die IAM für Sie generiert hat, indem Sie diese Informationen entweder anzeigen, kopieren und dann in eine sichere Datei auf Ihrem lokalen Computer einfügen oder indem Sie Anmeldeinformationen herunterladen wählen, um diese Informationen als CSV-Datei herunterzuladen. Sie benötigen diese Informationen, um eine Verbindung herzustellen. CodeCommit
2. Starten Sie eine Sitzung mit AppStream 2.0 auf der Grundlage der URL, die Sie in der Willkommens-E-Mail erhalten haben. Verwenden Sie Ihre E-Mail-Adresse als Benutzernamen und erstellen Sie Ihr Passwort.

3. Wählen Sie Ihren Enterprise Developer-Stack aus.
4. Wählen Sie auf der Menüseite Desktop aus, um zum Windows-Desktop zu gelangen, der von der Flotte gestreamt wird.

## Schritt 2: Erstellen Sie den Amazon S3-basierten virtuellen Ordner unter Windows (optional)

Wenn Rclone benötigt wird (siehe oben), erstellen Sie den Amazon S3-basierten virtuellen Ordner unter Windows: (optional, wenn alle Anwendungsartefakte ausschließlich aus Access stammen).  
CodeCommit

### Note

Wenn Sie Rclone bereits in der Vorschauversion der AWS Mainframe-Modernisierung verwendet haben, müssen Sie auf die neuere Version aktualisieren, die `m2-rclone.cmd` sich unter befindet. `C:\Users\Public`

1. Kopieren Sie die bereitgestellten `m2-rclone.cmd` Dateien `m2-rclone.conf` und mit `C:\Users\Public` dem `C:\Users\PhotonUser\My Files\Home Folder` Datei-Explorer in Ihren Home-Ordner.
2. Aktualisieren Sie die `m2-rclone.conf` Konfigurationsparameter mit Ihrem AWS Zugriffsschlüssel und dem entsprechenden Geheimnis sowie Ihrem AWS-Region.

```
[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. Nehmen Sie in `m2-rclone.cmd` die folgenden Änderungen vor:
  - Wechseln Sie `your-s3-bucket` zu Ihrem Amazon S3 S3-Bucket-Namen. z. B. `m2-s3-mybucket`.

- Wechseln Sie `your-s3-folder-key` zu Ihrem Amazon S3 S3-Bucket-Key. z. B. `myProject`.
- `your-local-folder-path` Wechseln Sie in den Pfad des Verzeichnisses, in das die Anwendungsdateien aus dem Amazon S3 S3-Bucket synchronisiert werden sollen, der sie enthält. z. B. `D:\PhotonUser\My Files\Home Folder\m2-new`. Dieses synchronisierte Verzeichnis muss ein Unterverzeichnis des Home-Ordners sein, damit AppStream 2.0 es beim Start und Ende der Sitzung ordnungsgemäß sichern und wiederherstellen kann.

```
:loop
timeout /T 10
"C:\Program Files\rclone\rclone.exe" sync m2-s3:your-s3-bucket/your-s3-folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:\PhotonUser\My Files\Home Folder\m2-rclone.conf"
goto :loop
```

4. Öffnen Sie eine Windows-Eingabeaufforderung, bei `C:\Users\PhotonUser\My Files\Home Folder` Bedarf `cd to`, und führen Sie den Befehl aus. `m2-rclone.cmd` Dieses Befehlskript führt eine Endlosschleife aus und synchronisiert Ihren Amazon S3 S3-Bucket und Ihren Schlüssel alle 10 Sekunden mit dem lokalen Ordner. Sie können das Timeout nach Bedarf anpassen. Sie sollten den Quellcode der Anwendung sehen, der sich im Amazon S3 S3-Bucket im Windows-Datei-Explorer befindet.

Um dem Set, an dem Sie gerade arbeiten, neue Dateien hinzuzufügen oder bestehende zu aktualisieren, laden Sie die Dateien in den Amazon S3 S3-Bucket hoch. Sie werden dann bei der nächsten unter definierten Iteration mit Ihrem Verzeichnis synchronisiert. `m2-rclone.cmd` Wenn Sie einige Dateien löschen möchten, löschen Sie sie ebenfalls aus dem Amazon S3 S3-Bucket. Beim nächsten Synchronisierungsvorgang werden sie aus Ihrem lokalen Verzeichnis gelöscht.

### Schritt 3: Klonen Sie das Repository

1. Navigieren Sie zum Anwendungsauswahlmenü in der oberen linken Ecke des Browserfensters und wählen Sie Enterprise Developer aus.
2. Schließe die von Enterprise Developer benötigte Workspace-Erstellung in deinem Home-Ordner ab, indem du `C:\Users\PhotonUser\My Files\Home Folder` (aka `D:\PhotonUser\My Files\Home Folder`) als Speicherort für den Workspace auswählst.



3. Klonen Sie in Enterprise Developer Ihr CodeCommit Repository, indem Sie zum Project Explorer gehen, mit der rechten Maustaste klicken und Import, Import..., Git, Projekte aus Git Clone URI wählen. Geben Sie dann Ihre CodeCommit spezifischen Anmeldedaten ein und füllen Sie den Eclipse-Dialog aus, um den Code zu importieren.

Das CodeCommit Git-Repository ist jetzt in deinem lokalen Workspace geklont.

Ihr Enterprise Developer-Workspace ist jetzt bereit, mit den Wartungsarbeiten an Ihrer Anwendung zu beginnen. Insbesondere können Sie die in Enterprise Developer integrierte lokale Instanz von Microfocus Enterprise Server (ES) verwenden, um Ihre Anwendung interaktiv zu debuggen und auszuführen, um Ihre Änderungen lokal zu validieren.

#### Note

Die lokale Enterprise Developer-Umgebung, einschließlich der lokalen Enterprise Server-Instanz, läuft unter Windows, während AWS Mainframe Modernization unter Linux ausgeführt wird. Wir empfehlen, dass Sie ergänzende Tests in der von AWS Mainframe Modernization bereitgestellten Linux-Umgebung durchführen, nachdem Sie die neue Anwendung für dieses Ziel übernommen CodeCommit und für dieses Ziel neu erstellt haben und bevor Sie die neue Anwendung in der Produktionsumgebung einführen.

## Nachfolgende Sitzungen

Wenn Sie einen Ordner auswählen, der unter AppStream 2.0-Verwaltung steht, z. B. den Home-Ordner für das Klonen Ihres CodeCommit Repositories, wird er sitzungsübergreifend transparent gespeichert und wiederhergestellt. Führen Sie die folgenden Schritte aus, wenn Sie das nächste Mal mit der Anwendung arbeiten müssen:

1. Starten Sie eine Sitzung mit AppStream 2.0 auf der Grundlage der URL, die Sie in der Willkommens-E-Mail erhalten haben.
2. Loggen Sie sich mit Ihrer E-Mail-Adresse und Ihrem permanenten Passwort ein.
3. Wählen Sie den Enterprise Developer-Stack aus.
4. Starten Sie `Rc1one`, um eine Verbindung zur Amazon S3-gestützten Festplatte herzustellen (siehe oben), wenn diese Option für die gemeinsame Nutzung der Workspace-Dateien verwendet wird.
5. Starten Sie Enterprise Developer, um Ihre Arbeit zu erledigen.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, damit sie Ihnen nicht weiter in Rechnung gestellt werden. Führen Sie folgende Schritte aus:

- Löschen Sie das CodeCommit Repository, das Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Löschen eines CodeCommit Repositorys](#).
- Löschen Sie die Datenbank, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen einer DB-Instance](#).

## Automatisierung für Micro Focus Enterprise Analyzer und Micro Focus Enterprise Developer Streaming-Sitzungen einrichten

Sie können beim Start und Ende der Sitzung automatisch ein Skript ausführen, um eine Automatisierung zu ermöglichen, die auf Ihren Kundenkontext zugeschnitten ist. Weitere Informationen zu dieser AppStream 2.0-Funktion finden Sie unter [Verwenden von Sitzungsskripten zur Verwaltung des Streaming-Erlebnisses Ihrer AppStream 2.0-Benutzer](#) im Amazon AppStream 2.0-Administrationshandbuch.

Für diese Funktion benötigen Sie mindestens die folgenden Versionen der Enterprise Analyzer- und Enterprise Developer-Images:

- `m2-enterprise-analyzer-v8.0.4.R1`
- `m2-enterprise-developer-v8.0.4.R1`

### Themen

- [Richten Sie die Automatisierung beim Sitzungsstart ein](#)
- [Richten Sie die Automatisierung am Ende der Sitzung ein](#)

## Richten Sie die Automatisierung beim Sitzungsstart ein

Wenn Sie ein Automatisierungsskript ausführen möchten, wenn Benutzer eine Verbindung zu AppStream 2.0 herstellen, erstellen Sie Ihr Skript und geben Sie ihm einen Namen `m2-user-setup.cmd`. Speichern Sie das Skript im AppStream 2.0-Home-Ordner für den Benutzer. Die

AppStream 2.0-Images, die AWS Mainframe Modernization bereitstellt, suchen an diesem Speicherort nach einem Skript mit diesem Namen und führen es aus, falls es existiert.

#### Note

Die Skriptdauer darf das in AppStream 2.0 festgelegte Limit, das derzeit 60 Sekunden beträgt, nicht überschreiten. Weitere Informationen finden Sie unter [Skripts ausführen, bevor Streaming-Sitzungen beginnen](#) im Amazon AppStream 2.0-Administrationshandbuch.

## Richten Sie die Automatisierung am Ende der Sitzung ein

Wenn Sie ein Automatisierungsskript ausführen möchten, wenn Benutzer die Verbindung zu AppStream 2.0 trennen, erstellen Sie Ihr Skript und geben Sie ihm einen Namen `m2-user-teardown.cmd`. Speichern Sie das Skript im AppStream 2.0-Home-Ordner für den Benutzer. Die AppStream 2.0-Images, die AWS Mainframe Modernization bereitstellt, suchen an diesem Speicherort nach einem Skript mit diesem Namen und führen es aus, falls es existiert.

#### Note

Die Skriptdauer darf das in AppStream 2.0 festgelegte Limit, das derzeit 60 Sekunden beträgt, nicht überschreiten. Weitere Informationen finden Sie unter [Skripts nach dem Ende von Streaming-Sitzungen ausführen](#) im Amazon AppStream 2.0-Administrationshandbuch.

## Datensätze als Tabellen und Spalten in Enterprise Developer anzeigen

Mit der Micro Focus-Laufzeit können Sie auf Mainframe-Datensätze zugreifen, die im Rahmen von AWS Mainframe Modernization bereitgestellt werden. Sie können die migrierten Datensätze als Tabellen und Spalten von einer Micro Focus Enterprise Developer-Instanz aus anzeigen. Wenn Sie Datensätze auf diese Weise anzeigen, können Sie:

- Führen Sie SQL `SELECT` Operationen mit den migrierten Datendateien durch.
- Machen Sie Daten außerhalb der migrierten Mainframe-Anwendung verfügbar, ohne die Anwendung zu ändern.
- Filtern Sie Daten einfach und speichern Sie sie als CSV oder in anderen Dateiformaten.

**Note**

Bei den Schritten 1 und 2 handelt es sich um einmalige Aktivitäten. Wiederholen Sie die Schritte 3 und 4 für jeden Datensatz, um die Datenbankansichten zu erstellen.

**Themen**

- [Voraussetzungen](#)
- [Schritt 1: ODBC-Verbindung zum Micro Focus-Datenspeicher \(Amazon RDS-Datenbank\) einrichten](#)
- [Schritt 2: Erstellen Sie die Datei MFDBFH.cfg](#)
- [Schritt 3: Erstellen Sie eine Strukturdatei \(STR\) für Ihr Copybook-Layout](#)
- [Schritt 4: Erstellen Sie eine Datenbankansicht mithilfe der Strukturdatei \(STR\)](#)
- [Schritt 5: Micro Focus-Datensätze als Tabellen und Spalten anzeigen](#)

**Voraussetzungen**

- Sie müssen über AppStream 2.0 Zugriff auf Micro Focus Enterprise Developer Desktop haben.
- Sie müssen eine Anwendung im Rahmen der AWS Mainframe-Modernisierung mithilfe der Micro Focus Runtime Engine bereitgestellt und ausgeführt haben.
- Sie speichern Ihre Anwendungsdaten in der Aurora PostgreSQL-kompatiblen Edition.

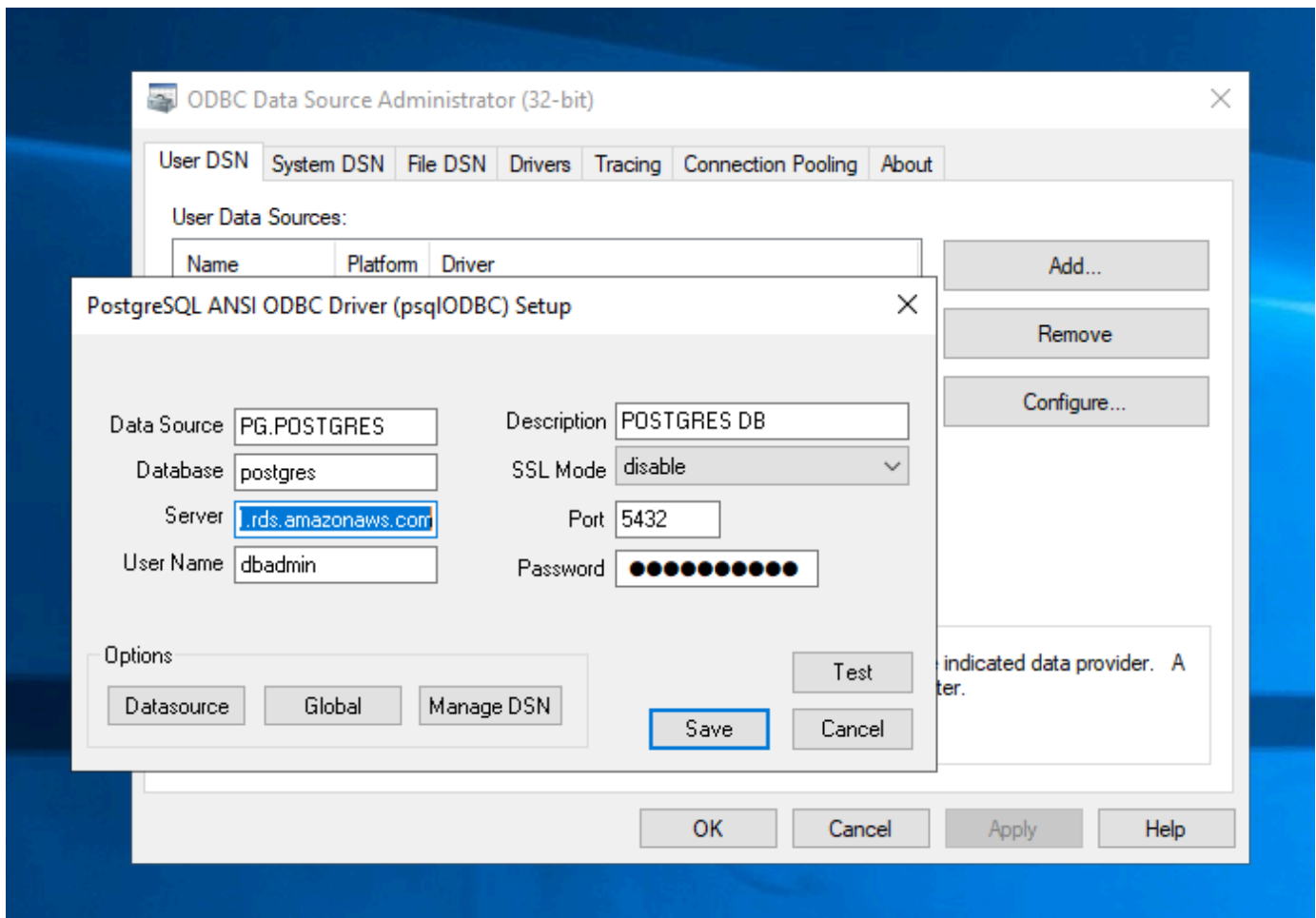
**Schritt 1: ODBC-Verbindung zum Micro Focus-Datenspeicher (Amazon RDS-Datenbank) einrichten**

In diesem Schritt richten Sie eine ODBC-Verbindung zu der Datenbank ein, die die Daten enthält, die Sie als Tabellen und Spalten anzeigen möchten. Dieser Schritt ist nur einmalig.

1. Melden Sie sich mit der AppStream 2.0-Streaming-URL bei Micro Focus Enterprise Developer Desktop an.
2. Öffnen Sie den ODBC-Datenquellenadministrator, wählen Sie Benutzer-DSN und dann Hinzufügen.
3. Wählen Sie unter Neue Datenquelle erstellen die Option PostgreSQL ANSI und dann Fertig stellen aus.

- Erstellen Sie eine Datenquelle für PG.POSTGRES indem Sie die erforderlichen Datenbankinformationen wie folgt angeben:

```
Data Source : PG.POSTGRES
Database    : postgres
Server      : rds_endpoint.rds.amazonaws.com
Port        : 5432
User Name   : user_name
Password    : user_password
```



- Wählen Sie Test, um sicherzustellen, dass die Verbindung funktioniert. Sie sollten die Meldung sehen, Connection successful ob der Test erfolgreich war.

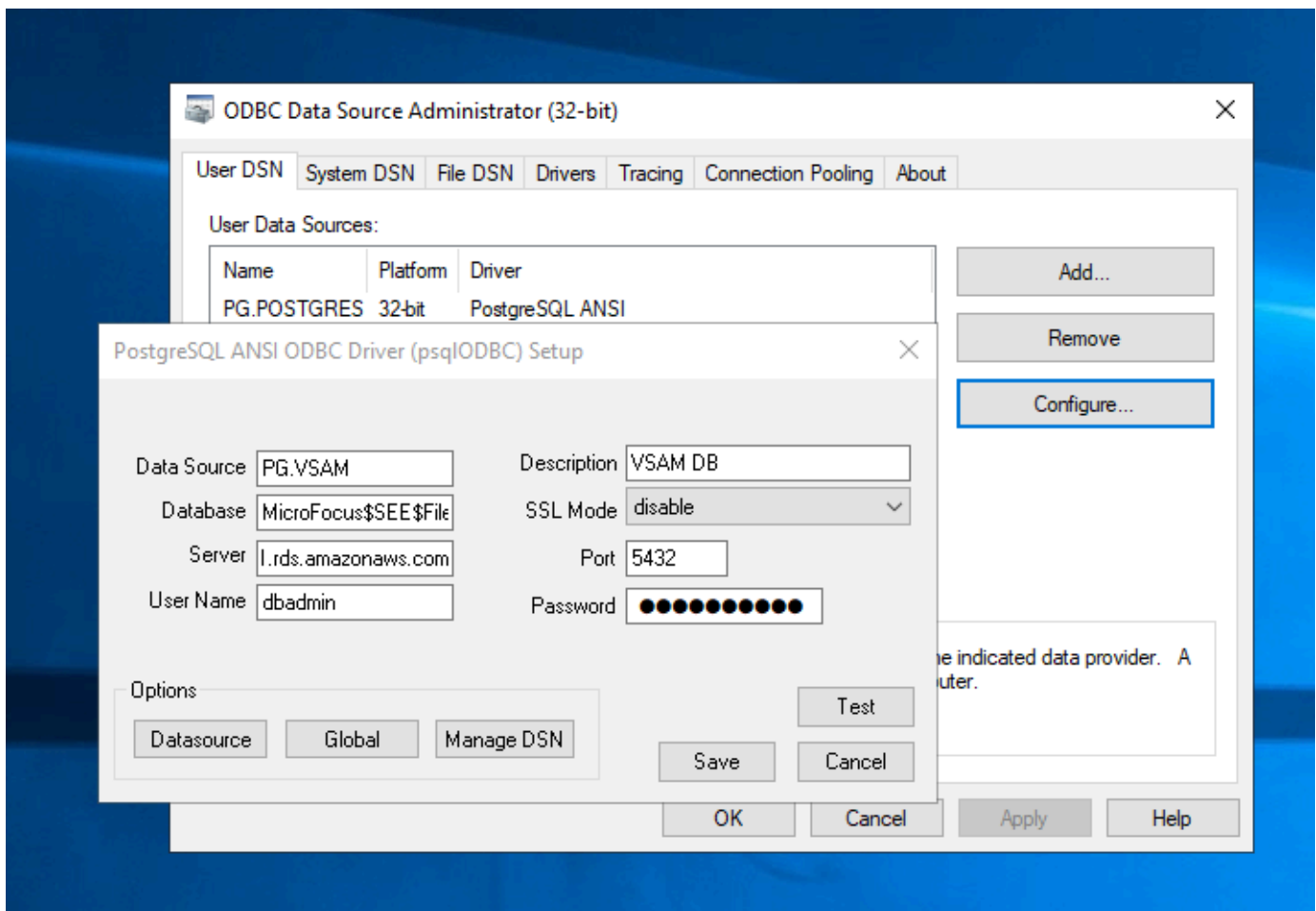
Wenn der Test nicht erfolgreich ist, überprüfen Sie die folgenden Informationen.

- [Fehlerbehebung für Amazon RDS](#)

- [Wie löse ich Probleme beim Herstellen einer Verbindung zu meiner Amazon RDS-DB-Instance?](#)

- Speichern Sie die Datenquelle.
- Erstellen Sie eine Datenquelle für PG.VSAM, testen Sie die Verbindung und speichern Sie die Datenquelle. Geben Sie die folgenden Datenbankinformationen an:

Data Source : PG.VSAM  
 Database : MicroFocus\$SEE\$Files\$VSAM  
 Server : *rds\_endpoint*.rds.amazonaws.com  
 Port : 5432  
 User Name : *user\_name*  
 Password : *user\_password*



## Schritt 2: Erstellen Sie die Datei MFDBFH.cfg

In diesem Schritt erstellen Sie eine Konfigurationsdatei, die den Micro Focus-Datenspeicher beschreibt. Dies ist ein einmaliger Konfigurationsschritt.

1. Erstellen Sie in Ihrem Home-Ordner, z. B. in D:\PhotonUser\My Files\Home Folder\MFED\cfg\MFDBFH.cfg, die Datei MFDBFH.cfg mit dem folgenden Inhalt.

```
<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. Überprüfen Sie die MFDBFH-Konfiguration, indem Sie die folgenden Befehle ausführen, um den Micro Focus-Datenspeicher abzufragen:

```
***
*** 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
```

## Schritt 3: Erstellen Sie eine Strukturdatei (STR) für Ihr Copybook-Layout

In diesem Schritt erstellen Sie eine Strukturdatei für Ihr Copybook-Layout, sodass Sie sie später verwenden können, um Datenbankansichten aus den Datensätzen zu erstellen.

1. Kompilieren Sie das Programm, das Ihrem Copybook zugeordnet ist. Wenn kein Programm das Copybook verwendet, erstellen und kompilieren Sie ein einfaches Programm wie das folgende mit einer COPY-Anweisung für Ihr Copybook.

```
IDENTIFICATION DIVISION.
    PROGRAM-ID. TESTPGM1.

    ENVIRONMENT DIVISION.
    CONFIGURATION SECTION.
```

```

DATA DIVISION.
WORKING-STORAGE SECTION.

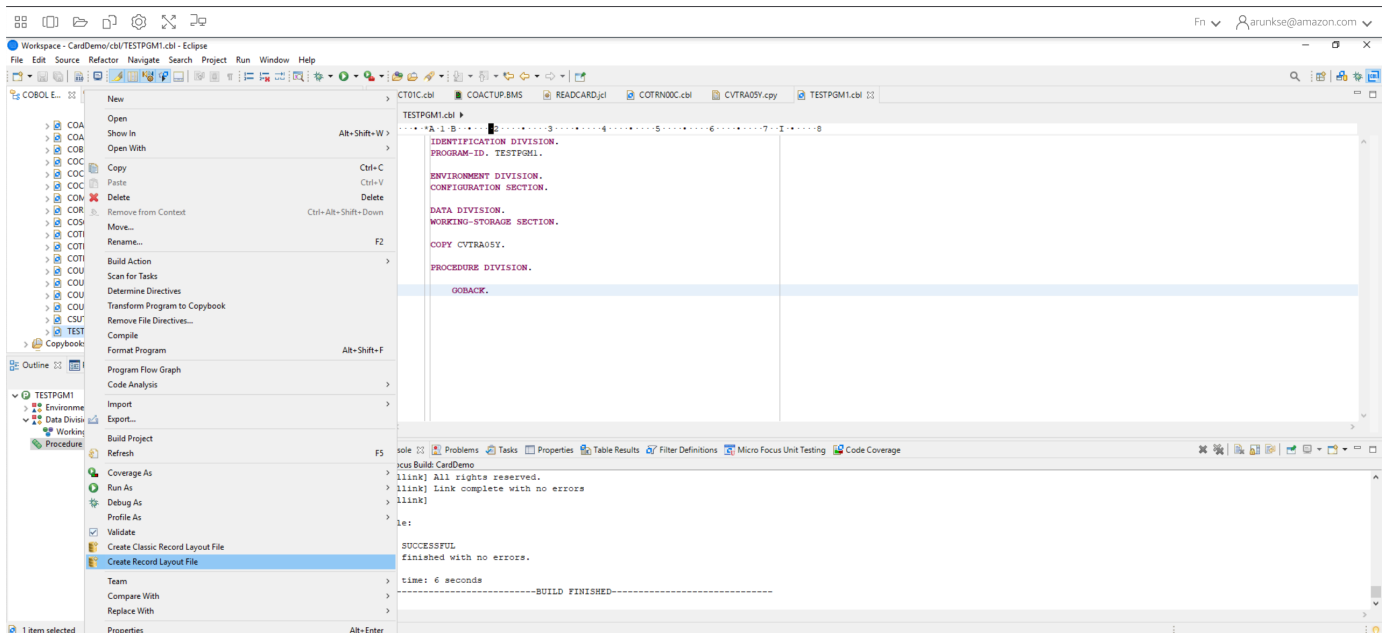
COPY CVTRA05Y.

PROCEDURE DIVISION.

GOBACK.

```

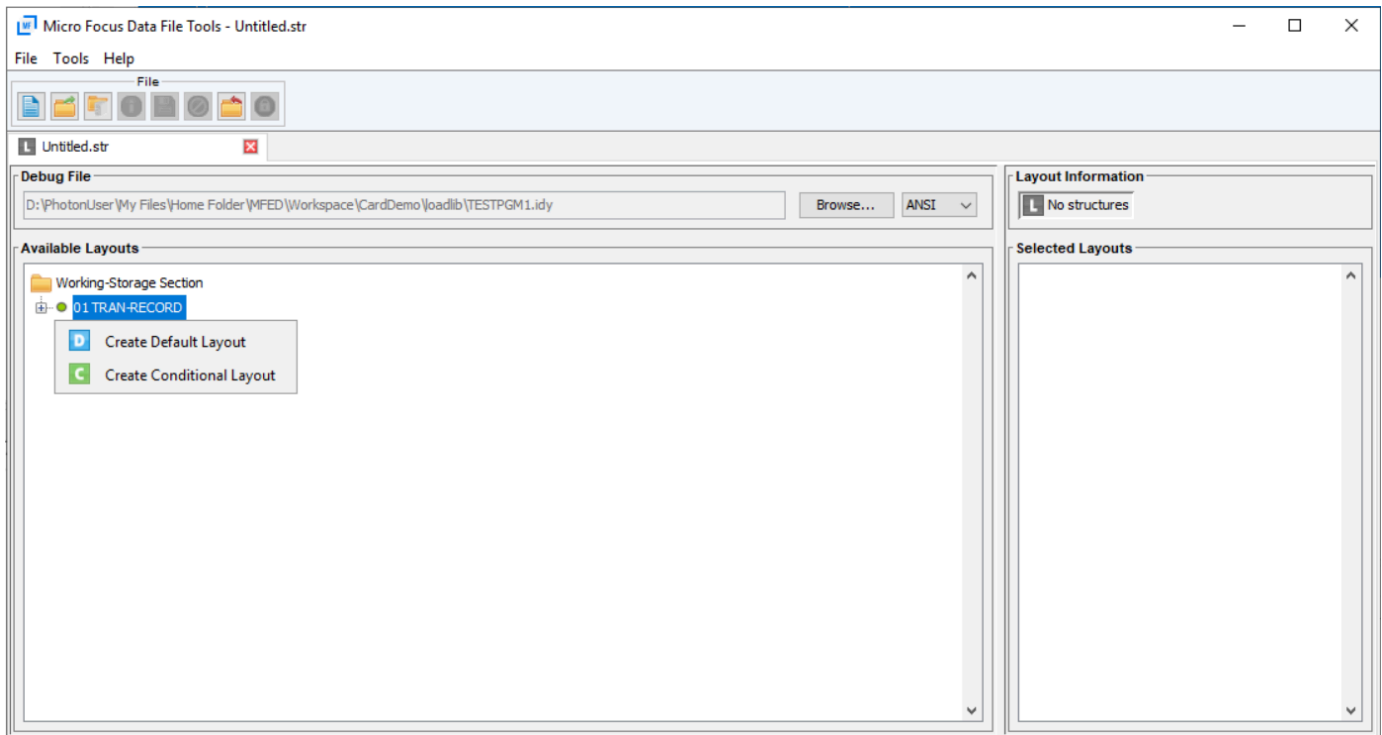
2. Klicken Sie nach erfolgreicher Kompilierung mit der rechten Maustaste auf das Programm und wählen Sie Create Record Layout File. Dadurch werden die Micro Focus Data File Tools mit der während der Kompilierung generierten IDY-Datei geöffnet.



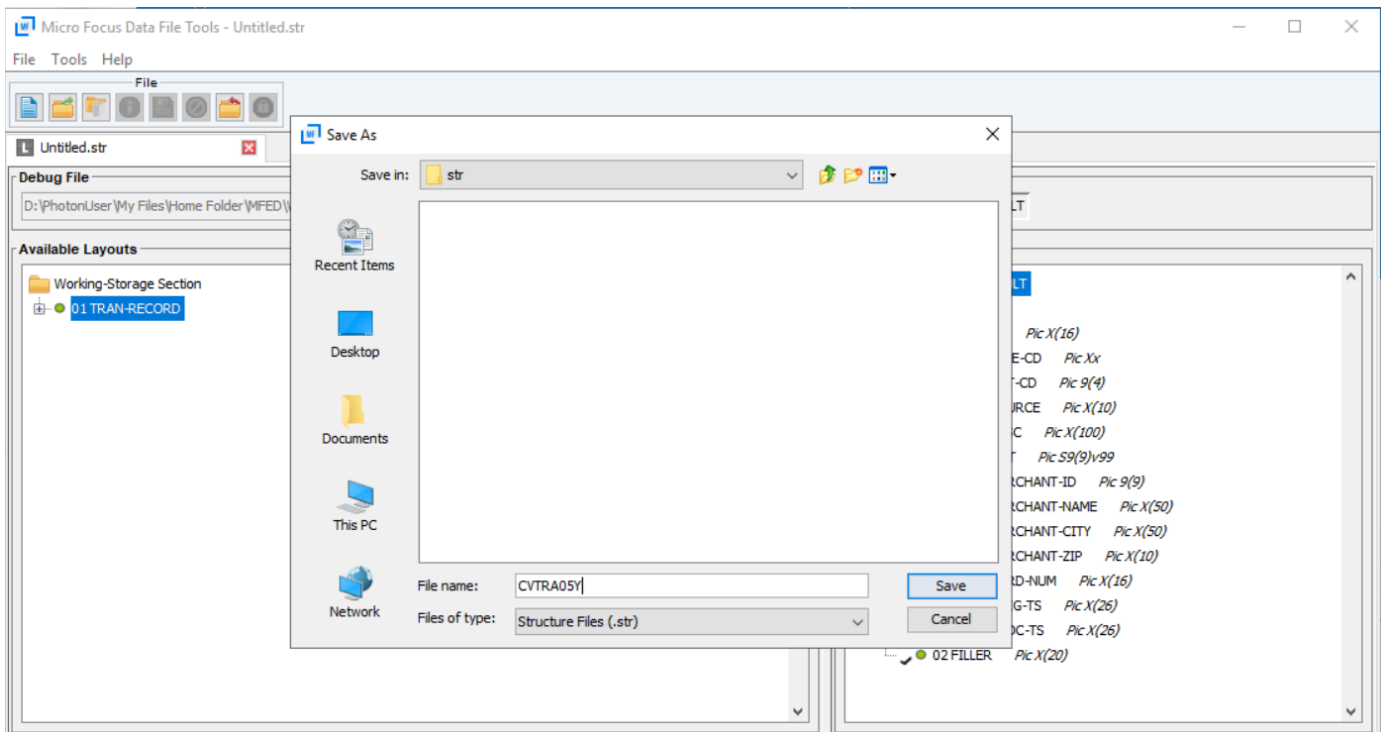
3. Klicken Sie mit der rechten Maustaste auf die Datensatzstruktur und wählen Sie je nach Layout Standardlayout erstellen (einzelne Struktur) oder Bedingtes Layout erstellen (mehrere Strukturen).

Weitere Informationen finden Sie unter [Strukturdateien und Layouts erstellen](#) in der Micro Focus-Dokumentation.





- Nachdem Sie das Layout erstellt haben, wählen Sie im Menü Datei und dann Speichern unter. Suchen und speichern Sie die Datei in Ihrem Home-Ordner mit demselben Dateinamen wie Ihr Copybook. Sie können wählen, ob Sie einen Ordner mit dem Namen erstellen `str` und alle Ihre Strukturdateien dort speichern möchten.



## Schritt 4: Erstellen Sie eine Datenbankansicht mithilfe der Strukturdatei (STR)

In diesem Schritt verwenden Sie die zuvor erstellte Strukturdatei, um eine Datenbankansicht für einen Datensatz zu erstellen.

- Verwenden Sie den `dbfhview` Befehl, um eine Datenbankansicht für einen Datensatz zu erstellen, der sich bereits im Micro Focus-Datenspeicher befindet, wie im folgenden Beispiel gezeigt.

```
##
    ## 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
```

## Schritt 5: Micro Focus-Datensätze als Tabellen und Spalten anzeigen

Stellen Sie in diesem Schritt eine Verbindung zur Datenbank her, pgAdmin sodass Sie Abfragen ausführen können, um die Datensätze wie Tabellen und Spalten anzuzeigen.

- Stellen Sie `MicroFocus$SEE$Files$VSAM` mit pgAdmin Connect zur Datenbank her und fragen Sie die Datenbankansicht ab, die Sie in Schritt 4 erstellt haben.

```
SELECT * FROM public."V_AWS.M2.CARDDEMO.TRANSACTION.VSAM.KSDS.DAT";
```

tran_id	tran_type_cd	tran_cat_cd	tran_source	tran_desc	tran_amt	tran_merchant_id	tran_merchant_name	tran_merchant_city	tran_merchant_zip	tran_card_num	tran_orig_id	
1	000000000683580	01	0001	POS TERM	Purchase at Abshire-Lowe	0000005...	800000000	Abshire-Lowe	North Enoshaven	72112	485945261287...	2022-06-10
2	0000000001774260	03	0001	OPERATOR	Return Item at Nitzsche, Nic...	0000009...	800000000	Nitzsche, Nicolas an...	Fidelshire	53378	092798710863...	2022-06-10
3	0000000006292564	01	0001	POS TERM	Purchase at Emser, Rob an...	0000000...	800000000	Emser, Rob and Gle...	North Makenziemo...	78487-7965	600961915067...	2022-06-10
4	0000000009101861	01	0001	POS TERM	Purchase at Guann LLC	0000002...	800000000	Guann LLC	South Lynn	51508-9166	804058041034...	2022-06-10
5	0000000010142252	01	0001	POS TERM	Purchase at Kertzmann-Scho...	0000004...	800000000	Kertzmann-Schoen	East Eulahstad	98754-1089	565683054498...	2022-06-10
6	0000000010229018	01	0001	POS TERM	Purchase at Gislason-Medhu...	0000008...	800000000	Gislason-Medhurst	Colleenburgh	23712-2080	737933563466...	2022-06-10
7	0000000016259484	03	0001	OPERATOR	Return Item at Sipes Inc	0000000...	800000000	Sipes Inc	Emilioside	93329	401150089177...	2022-06-10
8	0000000017874199	01	0001	POS TERM	Purchase at Legros Group	0000003...	800000000	Legros Group	Carmelborough	34849-5127	804058041034...	2022-06-10
9	0000000019065428	03	0001	OPERATOR	Return Item at Turcotte Group	0000005...	800000000	Turcotte Group	Andrewfurt	41346-3789	650335318179...	2022-06-10
10	0000000021711604	01	0001	POS TERM	Purchase at Gleason, Shana...	0000004...	800000000	Gleason, Shanahan a...	Myrticeport	21768-0823	950173372142...	2022-06-10
11	0000000025430891	01	0001	POS TERM	Purchase at Beatty-Hessel	0000000...	800000000	Beatty-Hessel	Simonisport	52595	326076361233...	2022-06-10
12	0000000028097268	01	0001	POS TERM	Purchase at Wolf, Cruickshank an...	0000002...	800000000	Wolf, Cruickshank an...	Fritzcheater	20195-5156	709414275105...	2022-06-10
13	0000000030752666	01	0001	POS TERM	Purchase at Rathe LLC	0000008...	800000000	Rathe LLC	Brendenfurt	35302-6495	376628198415...	2022-06-10
14	0000000032979555	01	0001	POS TERM	Purchase at Treutel-Leffler	0000000...	800000000	Treutel-Leffler	New Nicolette	65014-0045	650923036255...	2022-06-10
15	0000000033688127	01	0001	POS TERM	Purchase at Schinner-Steuher	0000009...	800000000	Schinner-Steuher	Schmittchester	50777-5535	376628198415...	2022-06-10
16	0000000040459589	01	0001	POS TERM	Purchase at Brekke, Bradlee and ...	0000007...	800000000	Brekke, Bradlee and ...	Veurmimouth	18481-5013	114216769287...	2022-06-10
17	000000004043636099	03	0001	OPERATOR	Return Item at Nader-Bayer	0000009...	800000000	Nader-Bayer	Goyetteville	35324	294013936230...	2022-06-10
18	0000000051205286	01	0001	POS TERM	Purchase at Goodwin, Von and Kr...	0000006...	800000000	Goodwin, Von and Kr...	Ericmouthe	03874	709414275105...	2022-06-10
19	0000000054288966	01	0001	POS TERM	Purchase at Cramin and Sons	0000005...	800000000	Cramin and Sons	Bartonside	08677	453478410771...	2022-06-10

## Tutorial: Verwenden von Vorlagen mit Micro Focus Enterprise Developer

In diesem Tutorial wird beschrieben, wie Sie Vorlagen und vordefinierte Projekte mit Micro Focus Enterprise Developer verwenden. Es deckt drei Anwendungsfälle ab. In allen Anwendungsfällen wird der im Beispiel enthaltene BankDemo Beispielcode verwendet. Um das Beispiel herunterzuladen, wählen Sie [bankdemo.zip](#).

### ⚠️ Important

Wenn Sie die Version von Enterprise Developer für Windows verwenden, können die vom Compiler generierten Binärdateien nur auf dem Enterprise Server ausgeführt werden, der im Lieferumfang von Enterprise Developer enthalten ist. Sie können sie nicht unter der AWS Mainframe Modernization Runtime ausführen, die auf Linux basiert.

### Themen

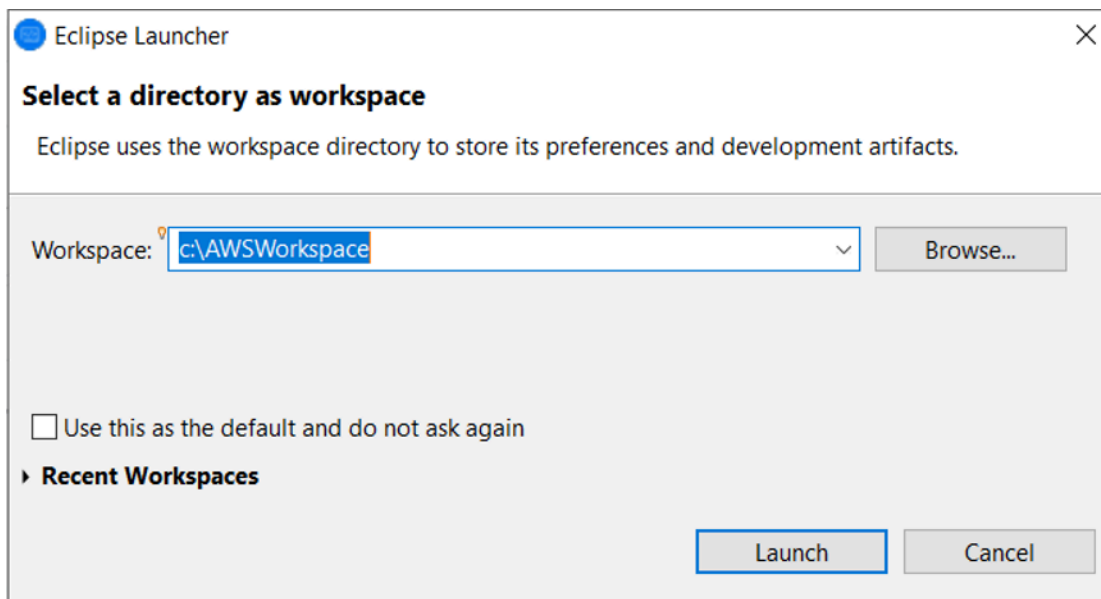
- [Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält](#)

- [Anwendungsfall 2 — Verwendung der COBOL-Projektvorlage ohne Quellkomponenten](#)
- [Anwendungsfall 3 — Verwendung des vordefinierten COBOL-Projekts, das auf die Quellordner verweist](#)
- [Verwenden der JSON-Vorlage für die Regionsdefinition](#)

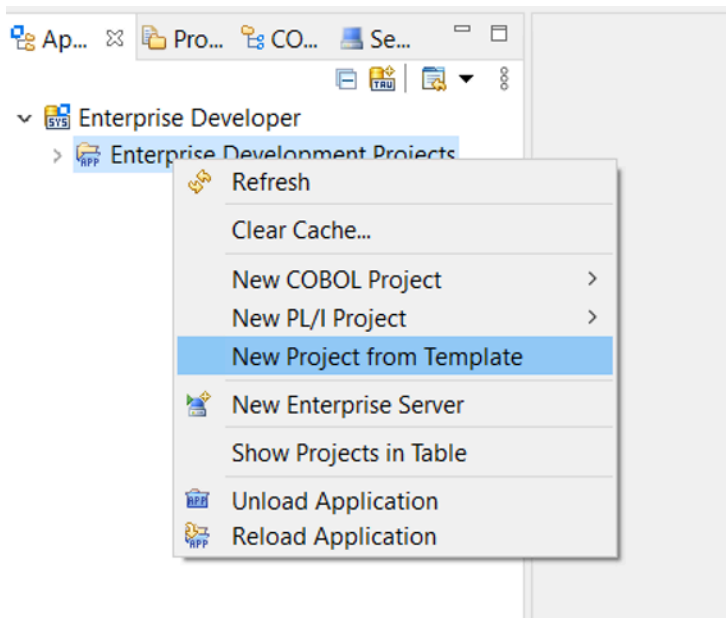
## Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält

Bei diesem Anwendungsfall müssen Sie die Quellkomponenten als Teil der Schritte vor der Einrichtung der Demo in die Verzeichnisstruktur des Templates kopieren. In der wurde [bankdemo.zip](#) dies gegenüber der ursprünglichen AWSTemplates.zip Lieferung geändert, um zu vermeiden, dass zwei Kopien der Quelle vorhanden sind.

1. Starten Sie Enterprise Developer und geben Sie den ausgewählten Workspace an.



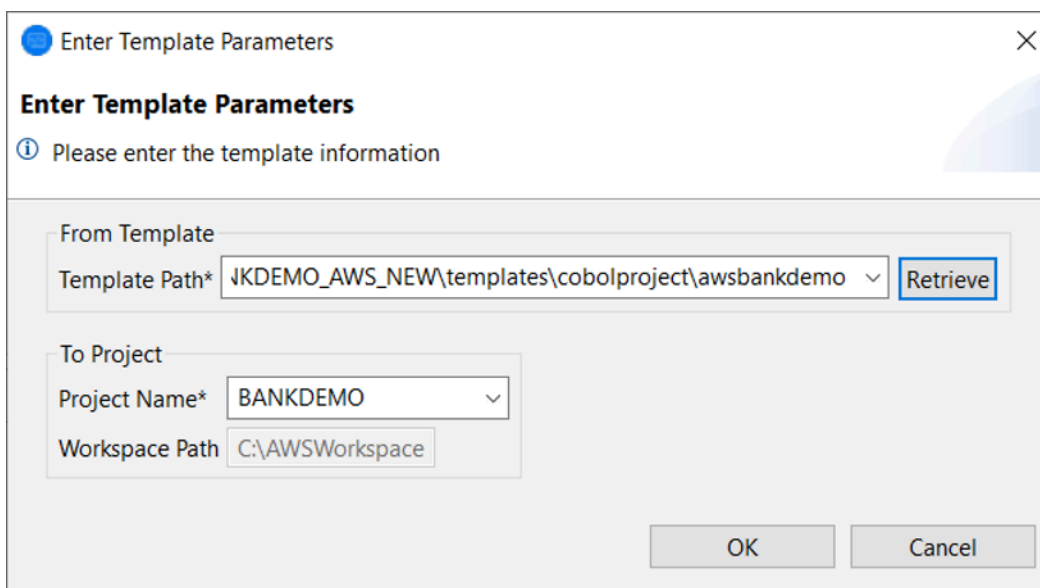
2. Wählen Sie in der Ansicht des Application Explorers in der Strukturansicht des Enterprise Development-Projekts im Kontextmenü die Option Neues Projekt aus Vorlage aus.



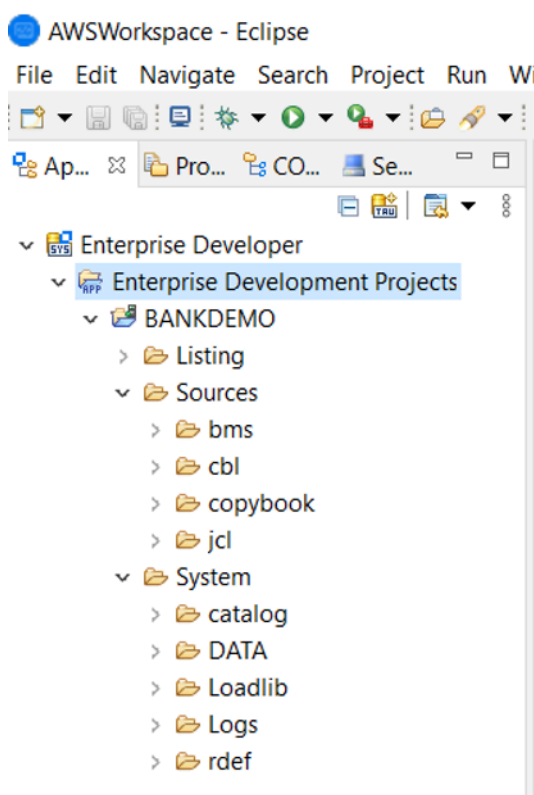
3. Geben Sie die Vorlagenparameter wie in der Abbildung gezeigt ein.

**Note**

Der Vorlagenpfad bezieht sich darauf, wo die ZIP-Datei extrahiert wurde.



4. Wenn Sie OK wählen, wird auf der Grundlage der bereitgestellten Vorlage ein lokales Eclipse-Entwicklungsprojekt mit einer vollständigen Quell- und Ausführungsumgebungsstruktur erstellt.



Die System Struktur enthält eine vollständige Ressourcendefinitionsdatei mit den erforderlichen Einträgen für BANKDEMO, den erforderlichen Katalog mit hinzugefügten Einträgen und den entsprechenden ASCII-Datendateien.

Da die Quellvorlagenstruktur alle Quellelemente enthält, werden diese Dateien in das lokale Projekt kopiert und daher automatisch in Enterprise Developer erstellt.

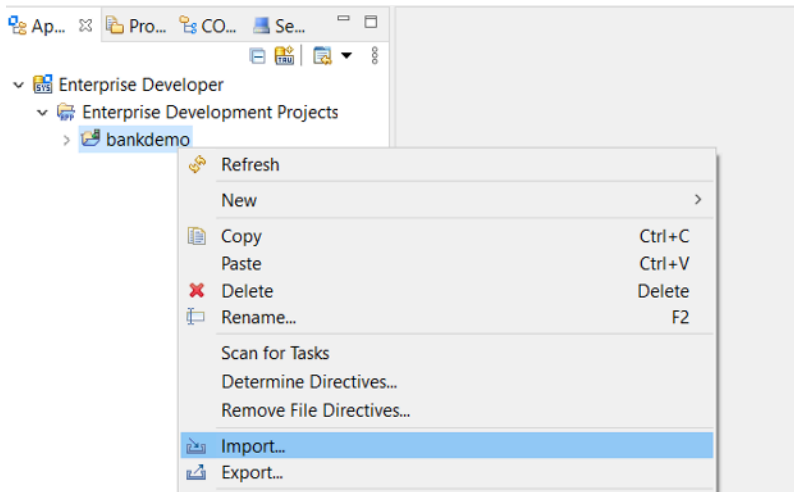
## Anwendungsfall 2 — Verwendung der COBOL-Projektvorlage ohne Quellkomponenten

Die Schritte 1 bis 3 sind identisch mit. [Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält](#)

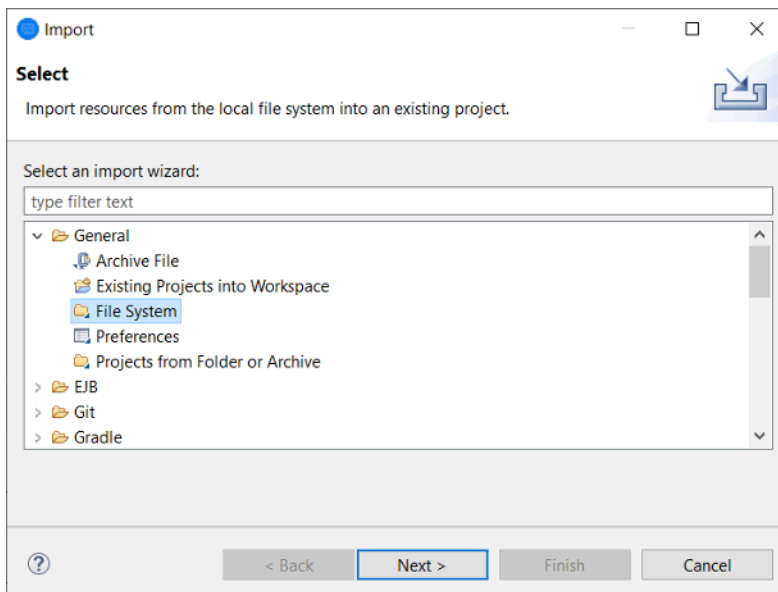
Die System Struktur in diesem Anwendungsfall enthält auch eine vollständige Ressourcendefinitionsdatei mit den erforderlichen Einträgen für BankDemo, den erforderlichen Katalog mit hinzugefügten Einträgen und den entsprechenden ASCII-Datendateien.

Die Quellstruktur der Vorlage enthält jedoch keine Komponenten. Sie müssen diese aus dem Quell-Repository, das Sie verwenden, in das Projekt importieren.

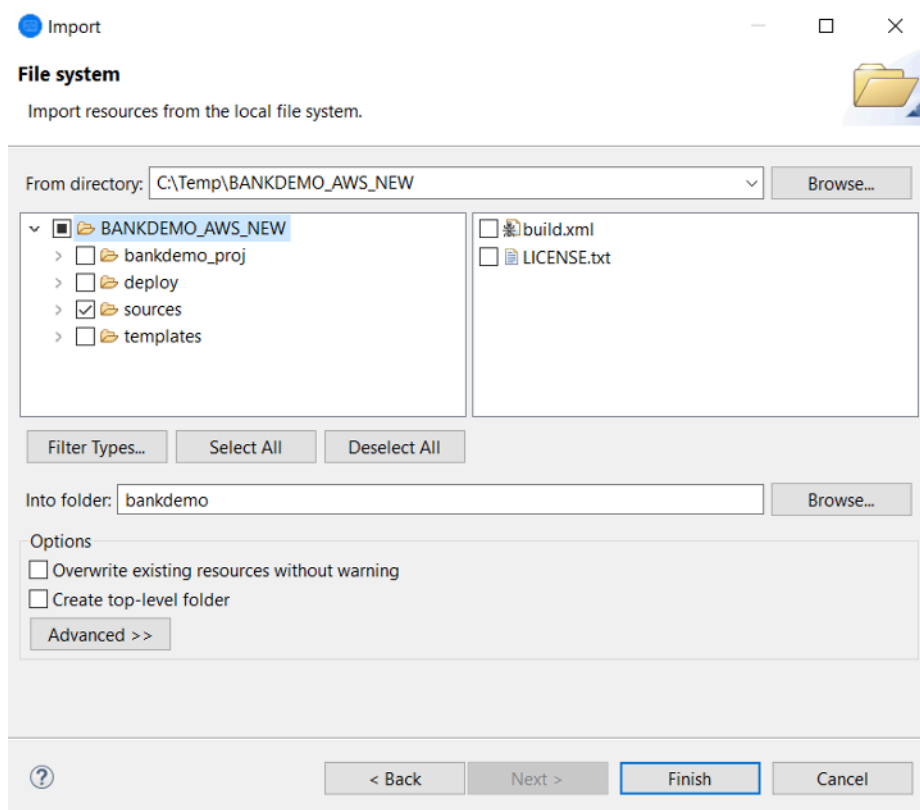
1. Wählen Sie den Projektnamen. Wählen Sie im entsprechenden Kontextmenü die Option Import.



2. Wählen Sie im daraufhin angezeigten Dialogfeld im Bereich Allgemein die Option Dateisystem und dann Weiter.



3. Füllen Sie das Feld Aus dem Verzeichnis aus, indem Sie das Dateisystem so durchsuchen, dass es auf den Repository-Ordner verweist. Wählen Sie alle Ordner aus, die Sie importieren möchten, z. B. sources Das Into folder Feld wird vorab ausgefüllt. Wählen Sie Finish (Abschließen).

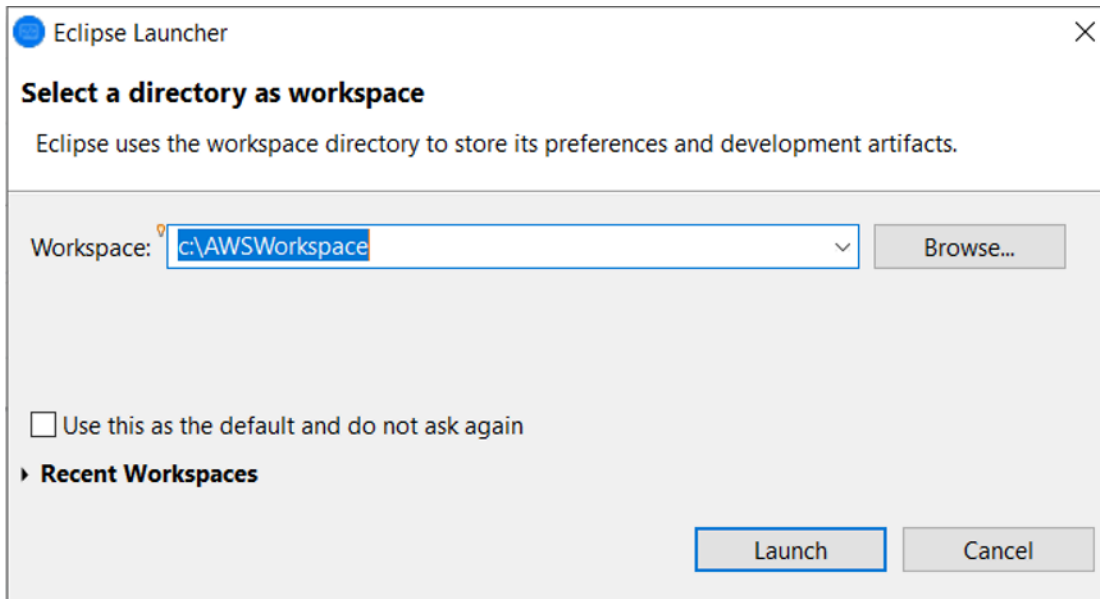


Nachdem die Quellvorlagenstruktur alle Quellelemente enthält, werden sie automatisch in Enterprise Developer erstellt.

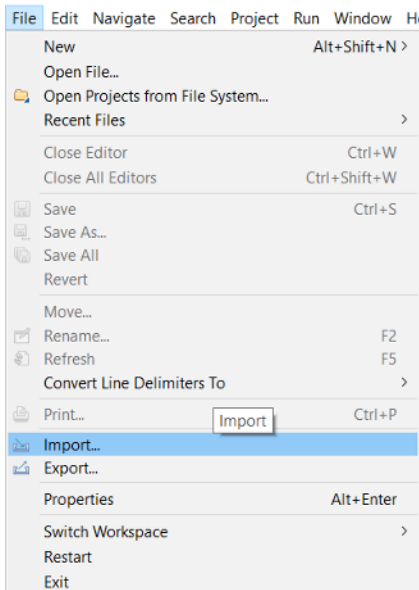
## Anwendungsfall 3 — Verwendung des vordefinierten COBOL-Projekts, das auf die Quellordner verweist

1. Starten Sie Enterprise Developer und geben Sie den ausgewählten Workspace an.

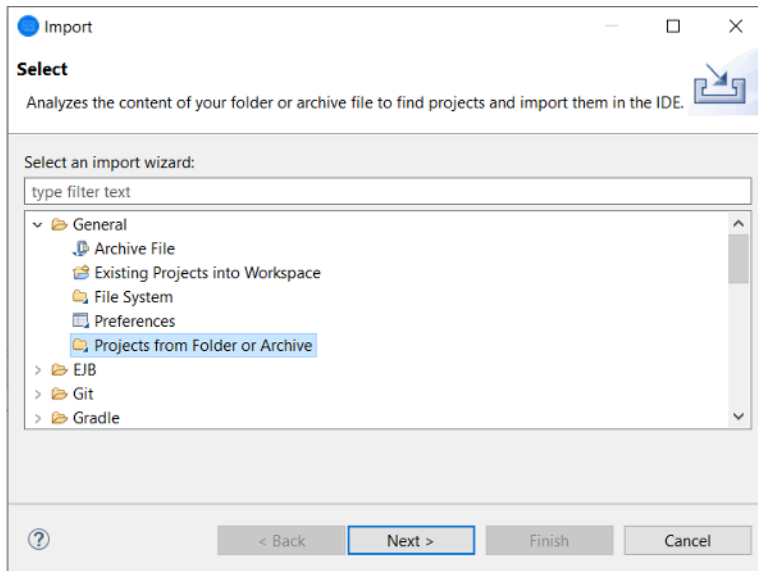




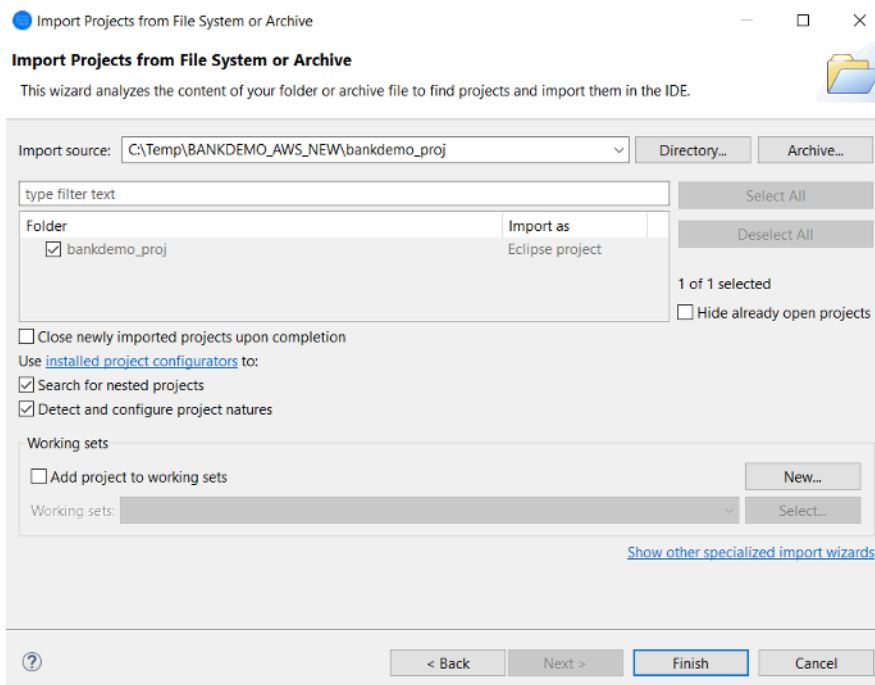
2. Wählen Sie im Menü Datei die Option Import aus.



3. Wählen Sie im daraufhin angezeigten Dialogfeld unter Allgemein die Option Projekte aus Ordner oder Archiv und dann Weiter.



- Füllen Sie die Importquelle aus, wählen Sie Verzeichnis und durchsuchen Sie das Dateisystem, um den vordefinierten Projektordner auszuwählen. Das darin enthaltene Projekt enthält Links zu den Quellordnern im selben Repository.

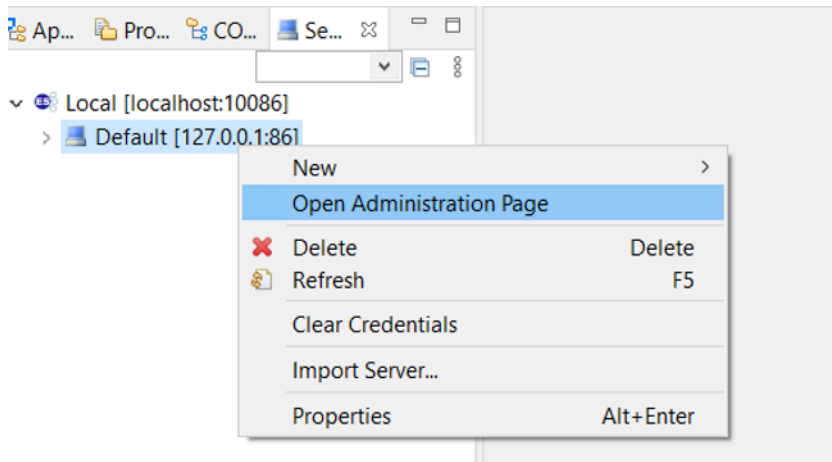


Wählen Sie Finish (Abschließen).

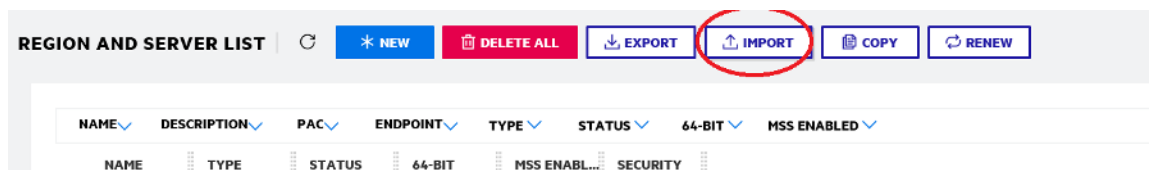
Da das Projekt durch die Links zum Quellordner aufgefüllt wird, wird der Code automatisch erstellt.

## Verwenden der JSON-Vorlage für die Regionsdefinition

1. Wechseln Sie zur Server Explorer-Ansicht. Wählen Sie im entsprechenden Kontextmenü die Option Administrationsseite öffnen, wodurch der Standardbrowser gestartet wird.



2. Wählen Sie im daraufhin angezeigten Fenster Enterprise Server Common Web Administration (ESCWA) die Option Import aus.



3. Wählen Sie den JSON-Importtyp und klicken Sie auf Weiter.

### CHOOSE IMPORT TYPE



#### JSON

Import a .json file by selecting a file on the host where the client browser is running.



#### XML

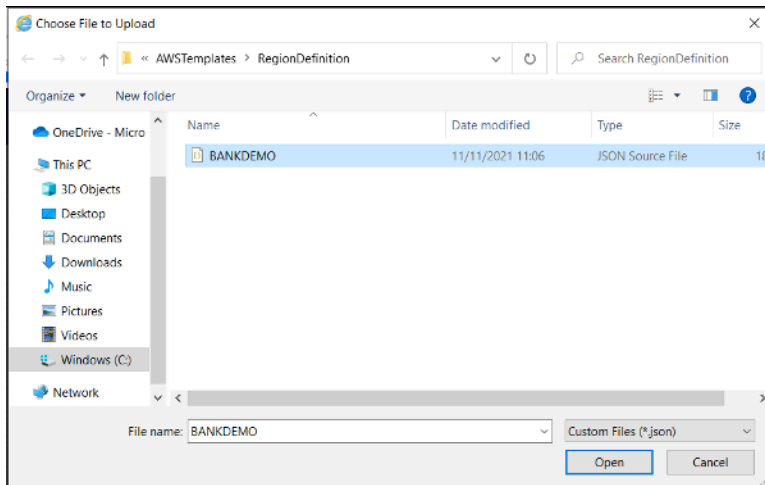
Import a .xml file by selecting a file on the host where the client browser is running.



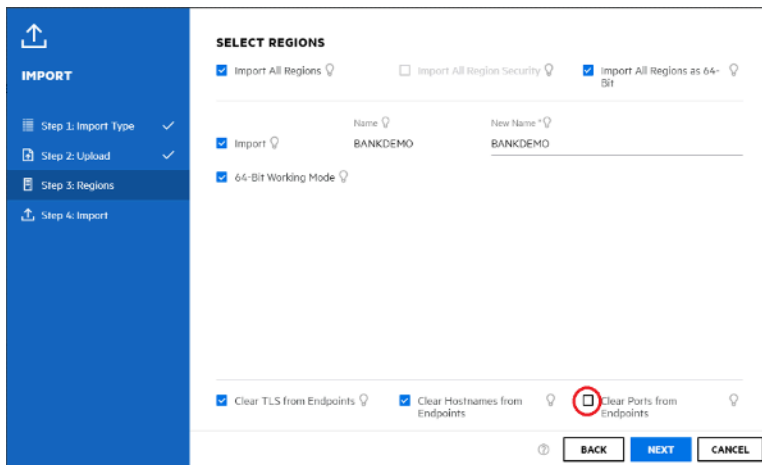
#### Legacy

Import a legacy repository (directory of .dat files) by selecting the directory location on the host where the Directory Server is running.

4. Laden Sie die bereitgestellte BANKDEMO .JSON Datei hoch.



Sobald Sie die Auswahl getroffen haben, wählen Sie Weiter.



Vergewissern Sie sich, dass im Bereich „Regionen auswählen“ die Option „Anschlüsse von Endpunkten löschen“ nicht aktiviert ist, und wählen Sie dann in den Bedienfeldern „Weiter“, bis der Bereich „Import durchführen“ angezeigt wird. Wählen Sie anschließend Import (Importieren) aus.



Klicken Sie abschließend auf Fertig stellen. Die BANKDEMO-Region wird dann der Serverliste hinzugefügt.

**REGION AND SERVER LIST** ↻ \* NEW 🗑️ DELETE ALL ⬇️ EXPORT ⬆️ IMPORT 📄 COPY 🔄 RENEW

NAME	DESCRIPTION	PAC	ENDPOINT	TYPE	STATUS	64-BIT	MSS ENABLED
BANKDEMO	Region				Stopped		✓
ESDEMO	Region				Stopped		
ESDEMO64	Region				Stopped	✓	

5. Navigieren Sie zu den Allgemeinen Eigenschaften für die Region BANKDEMO.
6. Scrollen Sie zum Abschnitt Konfiguration.
7. Die ESP-Umgebungsvariable muss auf den System Ordner gesetzt werden, der für das in den vorherigen Schritten erstellte Eclipse-Projekt relevant ist. Das sollte sein `workspacefolder/projectname/System`.

ADDITIONAL

```

Configuration Information ⓘ
[ES-Environment]
ESP={Enter Project System Folder Here}
MF_CHARSET=A
EXTFH=$ESP/EXTFH.cfg

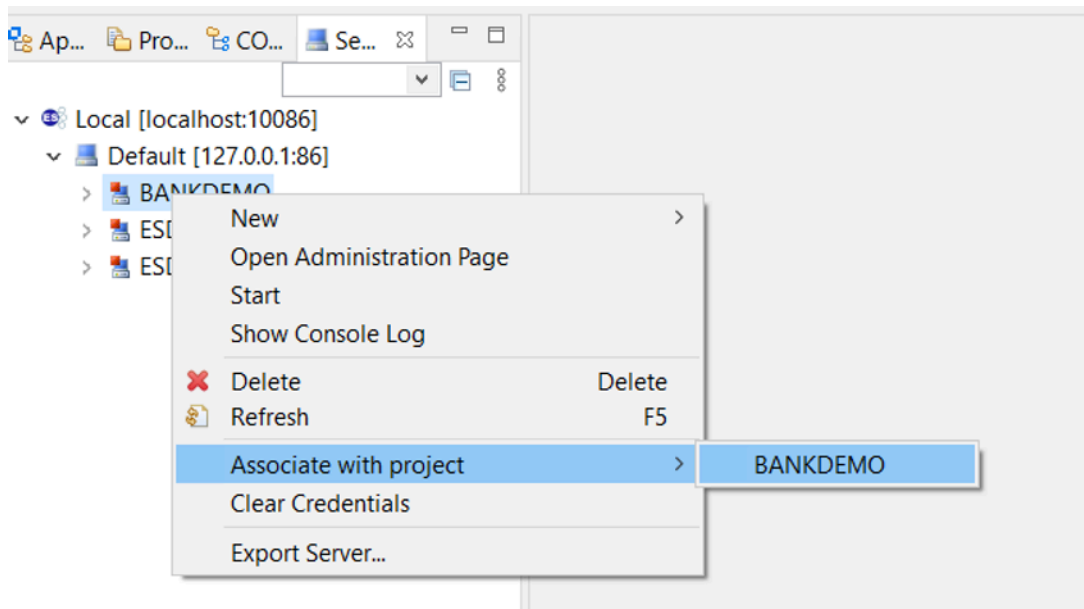
```

8. Klicken Sie auf Apply (Anwenden).

**APPLY**

Die Region ist jetzt vollständig für die Ausführung in Verbindung mit dem Eclipse COBOL-Projekt konfiguriert.

9. Verknüpfen Sie abschließend wieder in Enterprise Developer die importierte Region mit dem Projekt.



Die Enterprise Developer-Umgebung ist jetzt einsatzbereit und verfügt über eine voll funktionsfähige Version von BankDemo. Sie können Code für die Region bearbeiten, kompilieren und debuggen.

#### Important

Wenn Sie die Version von Enterprise Developer für Windows verwenden, können die vom Compiler generierten Binärdateien nur auf dem Enterprise Server ausgeführt werden, der im Lieferumfang von Enterprise Developer enthalten ist. Sie können sie nicht unter der AWS Mainframe Modernization Runtime ausführen, die auf Linux basiert.

## Tutorial: Den Micro Focus-Build für die BankDemo Beispielanwendung einrichten

AWS Die Mainframe-Modernisierung bietet Ihnen die Möglichkeit, Builds und CI/CD-Pipelines (Continuous Integration/Continuous Delivery) für Ihre migrierten Anwendungen einzurichten. Diese Builds und Pipelines nutzen AWS CodeBuild und stellen diese Funktionen bereit. AWS CodeCommit AWS CodePipeline CodeBuild ist ein vollständig verwalteter Build-Service, der Ihren Quellcode kompiliert, Komponententests durchführt und Artefakte erzeugt, die sofort einsatzbereit sind. CodeCommit ist ein Versionskontrolldienst, mit dem Sie Git-Repositorys privat in der Cloud speichern und verwalten können. AWS CodePipeline ist ein Dienst zur kontinuierlichen Bereitstellung, mit dem

Sie die zur Veröffentlichung Ihrer Software erforderlichen Schritte modellieren, visualisieren und automatisieren können.

Dieses Tutorial zeigt, wie Sie AWS CodeBuild den Quellcode der BankDemo Beispielanwendung aus Amazon S3 kompilieren und den kompilierten Code anschließend zurück nach Amazon S3 exportieren.

AWS CodeBuild ist ein vollständig verwalteter Continuous Integration-Service, der Quellcode kompiliert, Tests durchführt und Softwarepakete erstellt, die sofort einsatzbereit sind. Mit CodeBuild können Sie vorgefertigte Build-Umgebungen verwenden oder benutzerdefinierte Build-Umgebungen erstellen, die Ihre eigenen Build-Tools verwenden. In diesem Demo-Szenario wird die zweite Option verwendet. Es besteht aus einer CodeBuild Build-Umgebung, die ein vorgefertigtes Docker-Image verwendet.

#### Important

Bevor Sie mit Ihrem Mainframe-Modernisierungsprojekt beginnen, empfehlen wir Ihnen, sich über das [AWS Migration Acceleration Program \(MAP\) für Mainframe zu informieren oder sich an AWS Mainframe-Spezialisten](#) zu wenden, um mehr über die zur Modernisierung einer Mainframe-Anwendung erforderlichen Schritte zu erfahren.

## Themen

- [Voraussetzungen](#)
- [Schritt 1: Amazon S3 S3-Buckets erstellen](#)
- [Schritt 2: Erstellen Sie die Build-Spezifikationsdatei](#)
- [Schritt 3: Laden Sie die Quelldateien hoch](#)
- [Schritt 4: IAM-Richtlinien erstellen](#)
- [Schritt 5: Erstellen Sie eine IAM-Rolle](#)
- [Schritt 6: Hängen Sie die IAM-Richtlinien an die IAM-Rolle an](#)
- [Schritt 7: Erstellen Sie das CodeBuild Projekt](#)
- [Schritt 8: Starten Sie den Build](#)
- [Schritt 9: Laden Sie die Ausgabeartefakte herunter](#)
- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

Bevor Sie mit diesem Tutorial beginnen, müssen Sie die folgenden Voraussetzungen erfüllen.

- Laden Sie die [BankDemo Beispielanwendung](#) herunter und entpacken Sie sie in einen Ordner. Der Quellordner enthält COBOL-Programme und Copybooks sowie CICS BMS-Definitionen. Er enthält auch einen JCL-Ordner als Referenz, obwohl Sie JCL nicht erstellen müssen. Der Ordner enthält auch die für den Build erforderlichen Metadateien.
- Wählen Sie in der AWS Mainframe Modernization Console die Option Tools aus. Wählen Sie unter Analysis, Development and Build Assets die Option Assets mit meinem AWS-Konto teilen aus.

### Schritt 1: Amazon S3 S3-Buckets erstellen

In diesem Schritt erstellen Sie zwei Amazon S3 S3-Buckets. Der erste ist ein Eingabe-Bucket für den Quellcode und der andere ist ein Ausgabe-Bucket für die Build-Ausgabe. Weitere Informationen finden Sie unter [Erstellen, Konfigurieren und Arbeiten mit Amazon S3 S3-Buckets](#) im Amazon S3 S3-Benutzerhandbuch.

1. Um den Eingabe-Bucket zu erstellen, melden Sie sich bei der Amazon S3 S3-Konsole an und wählen Sie Create Bucket.
2. Geben Sie in der allgemeinen Konfiguration einen Namen für den Bucket an und geben Sie an AWS-Region , wo Sie den Bucket erstellen möchten. Ein Beispiel für den Namen ist `codebuild-regionId-accountId-input-bucket` Where `regionId` is the AWS-Region of the Bucket und `accountId` ist Ihre AWS-Konto ID.

#### Note

Wenn Sie den Bucket in einer anderen Region als USA AWS-Region Ost (Nord-Virginia) erstellen, geben Sie den `LocationConstraint` Parameter an. Weitere Informationen finden Sie unter [Create Bucket](#) in der Amazon Simple Storage Service API-Referenz.

3. Behalten Sie alle anderen Einstellungen bei und wählen Sie Create Bucket.
4. Wiederholen Sie die Schritte 1—3, um den Ausgabe-Bucket zu erstellen. Ein Beispiel für den Namen ist `codebuild-regionId-accountId-output-bucket` where `regionId` is the AWS-Region of the Bucket und `accountId` ist Ihre AWS-Konto ID.



Welche Namen Sie auch immer für diese Buckets wählen, achten Sie darauf, sie in diesem Tutorial zu verwenden.

## Schritt 2: Erstellen Sie die Build-Spezifikationsdatei

In diesem Schritt erstellen Sie eine Build-Spezifikationsdatei. Diese Datei enthält Build-Befehle und zugehörige Einstellungen im YAML-Format, CodeBuild um den Build auszuführen. Weitere Informationen finden Sie unter [Referenz zur Build-Spezifikation für CodeBuild](#) im AWS CodeBuild Benutzerhandbuch.

1. Erstellen Sie als Voraussetzung eine Datei mit dem Namen `buildspec.yml` in dem Verzeichnis, das Sie entpackt haben.
2. Fügen Sie der Datei den folgenden Inhalt hinzu und speichern Sie sie. Für diese Datei sind keine Änderungen erforderlich.

```
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/**
```

Hier `$CODEBUILD_SRC_DIR/target` sind Umgebungsvariablen `CODEBUILD_BUILD_ID`, `CODEBUILD_BUILD_ARN`, `$CODEBUILD_SRC_DIR/source`, und innerhalb verfügbar CodeBuild. Weitere Informationen finden Sie unter [Umgebungsvariablen in Build-Umgebungen](#).

Zu diesem Zeitpunkt sollte Ihr Verzeichnis so aussehen.

```
(root directory name)
|-- build.xml
|-- buildspec.yml
|-- LICENSE.txt
|-- source
    |... etc.
```

3. Komprimieren Sie den Inhalt des Ordners in eine Datei mit dem Namen `BankDemo.zip`. Für dieses Tutorial können Sie den Ordner nicht komprimieren. Stattdessen komprimieren Sie den Inhalt des Ordners in der Datei `BankDemo.zip`.

## Schritt 3: Laden Sie die Quelldateien hoch

In diesem Schritt laden Sie den Quellcode für die BankDemo Beispielanwendung in Ihren Amazon S3 S3-Eingabe-Bucket hoch.

1. Melden Sie sich bei der Amazon S3 S3-Konsole an und wählen Sie im linken Navigationsbereich Buckets aus. Wählen Sie dann den Eingabe-Bucket aus, den Sie zuvor erstellt haben.
2. Wählen Sie unter Objekte die Option Hochladen aus.
3. Wählen Sie im Bereich Dateien und Ordner die Option Dateien hinzufügen aus.
4. Navigieren Sie zu Ihrer `BankDemo.zip` Datei und wählen Sie sie aus.
5. Klicken Sie auf Hochladen.

## Schritt 4: IAM-Richtlinien erstellen

In diesem Schritt erstellen Sie zwei [IAM-Richtlinien](#). Eine Richtlinie gewährt AWS Mainframe Modernization die Erlaubnis, auf das Docker-Image zuzugreifen und es zu verwenden, das die Micro Focus-Build-Tools enthält. Diese Richtlinie ist nicht auf Kunden zugeschnitten. Die andere Richtlinie

gewährt AWS Mainframe Modernization die Erlaubnis, mit den Eingabe- und Ausgabe-Buckets sowie mit den generierten [CloudWatch Amazon-Protokollen](#) zu interagieren. CodeBuild

Weitere Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter [Bearbeiten von IAM-Richtlinien im IAM-Benutzerhandbuch](#).

So erstellen Sie eine Richtlinie für den Zugriff auf Docker-Images

1. Kopieren Sie in der IAM-Konsole das folgende Richtliniendokument und fügen Sie es in den Richtlinien-Editor ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "arn:aws:ecr:*:673918848628:repository/m2-enterprise-build-
tools"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::aws-m2-repo-*/*"
    }
  ]
}
```

2. Geben Sie einen Namen für die Richtlinie ein, z. B. m2CodeBuildPolicy

Um eine Richtlinie zu erstellen, die es AWS Mainframe Modernization ermöglicht, mit Buckets und Logs zu interagieren

1. Kopieren Sie in der IAM-Konsole das folgende Richtliniendokument und fügen Sie es in den Policy-Editor ein. Stellen Sie sicher, dass Sie `regionId` auf die AWS-Region und `accountId` auf Ihre AWS-Konto aktualisieren.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/codebuild-bankdemo-project",
        "arn:aws:logs:regionId:accountId:log-group:/aws/codebuild/codebuild-bankdemo-project:*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::codebuild-regionId-accountId-input-bucket",
        "arn:aws:s3:::codebuild-regionId-accountId-input-bucket/*",
        "arn:aws:s3:::codebuild-regionId-accountId-output-bucket",
        "arn:aws:s3:::codebuild-regionId-accountId-output-bucket/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

2. Geben Sie einen Namen für die Richtlinie ein, BankdemoCodeBuildRolePolicy z. B.

## Schritt 5: Erstellen Sie eine IAM-Rolle

In diesem Schritt erstellen Sie eine neue [IAM-Rolle](#), mit der Sie für Sie mit AWS Ressourcen interagieren können CodeBuild , nachdem Sie die zuvor erstellten IAM-Richtlinien mit dieser neuen IAM-Rolle verknüpft haben.

Informationen zum Erstellen einer Servicerolle finden Sie unter [Creating a Role to Delegate Permissions to an AWS Service](#) im IAM-Benutzerhandbuch,.

1. Melden Sie sich bei der IAM-Konsole an und wählen Sie im linken Navigationsbereich Rollen aus.
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie unter Vertrauenswürdiger Entitätstyp die Option AWS-Service aus.
4. Wählen Sie unter Anwendungsfälle für andere AWS-Services die Option aus CodeBuild, und wählen Sie dann CodeBuilderneut.
5. Wählen Sie Next (Weiter).
6. Wählen Sie auf der Seite Add permissions (Berechtigungen hinzufügen) die Option Next (Weiter) aus. Sie weisen der Rolle später eine Richtlinie zu.
7. Geben Sie unter Rollendetails einen Namen für die Rolle ein, BankdemoCodeBuildServiceRole z. B.
8. Stellen Sie unter Vertrauenswürdige Entitäten auswählen sicher, dass das Richtliniendokument wie folgt aussieht:

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

```
}
```

9. Wählen Sie Rolle erstellen aus.

## Schritt 6: Hängen Sie die IAM-Richtlinien an die IAM-Rolle an

In diesem Schritt fügen Sie die beiden IAM-Richtlinien, die Sie zuvor erstellt haben, der `BankdemoCodeBuildServiceRole` IAM-Rolle hinzu.

1. Melden Sie sich bei der IAM-Konsole an und wählen Sie im linken Navigationsbereich Rollen aus.
2. Wählen Sie unter Rollen die Rolle aus, die Sie zuvor erstellt haben, `BankdemoCodeBuildServiceRole` z. B.
3. Wählen Sie unter Berechtigungsrichtlinien die Option Berechtigungen hinzufügen und dann Richtlinien anhängen aus.
4. Wählen Sie unter Andere Berechtigungsrichtlinien die Richtlinien aus, die Sie zuvor erstellt haben, z. B. `m2CodeBuildPolicy` und `BankdemoCodeBuildRolePolicy`.
5. Wählen Sie Richtlinien anfügen.

## Schritt 7: Erstellen Sie das CodeBuild Projekt

In diesem Schritt erstellen Sie das CodeBuild Projekt.

1. Melden Sie sich bei der CodeBuild Konsole an und wählen Sie Build-Projekt erstellen.
2. Geben Sie im Abschnitt Projektkonfiguration einen Namen für das Projekt ein, `codebuild-bankdemo-project` z. B.
3. Wählen Sie im Abschnitt Quelle für Quellanbieter Amazon S3 und dann den Eingabe-Bucket aus, den Sie zuvor erstellt haben, zum Beispiel `codebuild-regionId-accountId-input-bucket`.
4. Geben Sie im Feld S3-Objektschlüssel oder S3-Ordner den Namen der ZIP-Datei ein, die Sie in den S3-Bucket hochgeladen haben. In diesem Fall lautet der Dateiname `bankdemo.zip`.
5. Wählen Sie im Bereich Umgebung die Option Benutzerdefiniertes Bild aus.
6. Wählen Sie im Feld Umgebungstyp die Option Linux aus.
7. Wählen Sie unter Image-Registrierung die Option Andere Registrierung aus.

## 8. Im Feld Externe Registrierungs-URL

- Für Micro Focus v9: Geben Sie ein `673918848628.dkr.ecr.us-west-1.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1`. Wenn Sie eine andere AWS Region mit Micro Focus v9 verwenden, können Sie auch angeben `673918848628.dkr.ecr.<m2-region>.amazonaws.com/m2-enterprise-build-tools:9.0.7.R1`, <m2-region>in welcher AWS Region der AWS Mainframe-Modernisierungsservice verfügbar ist (z. B.). `eu-west-3`
  - Für Micro Focus v8: Geben Sie ein `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:8.0.9.R1`
  - Für Micro Focus v7: Geben Sie ein `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:7.0.R10`
9. Wählen Sie unter Servicerolle die Option Bestehende Servicerolle und wählen Sie im Feld Rollen-ARN die Servicerolle aus, die Sie zuvor erstellt haben, z. B. `BankdemoCodeBuildServiceRole`.
  10. Wählen Sie im Abschnitt Buildspec die Option Buildspec-Datei verwenden aus.
  11. Wählen Sie im Abschnitt Artifacts unter Type Amazon S3 und dann Ihren Output-Bucket aus, zum Beispiel `codebuild-regionId-accountId-output-bucket`.
  12. Geben Sie im Feld Name den Namen eines Ordners im Bucket ein, der die Build-Ausgabeartefakte enthalten soll, `bankdemo-output.zip` z. B.
  13. Wählen Sie unter Verpackung von Artefakten die Option Zip aus.
  14. Wählen Sie Create build project (Build-Projekt erstellen) aus.

## Schritt 8: Starten Sie den Build

In diesem Schritt starten Sie den Build.

1. Loggen Sie sich in die CodeBuild Konsole ein.
2. Wählen Sie im linken Navigationsbereich die Option Projekte erstellen aus.
3. Wählen Sie das Build-Projekt aus, das Sie zuvor erstellt haben, `codebuild-bankdemo-project` z. B.
4. Wählen Sie Start build (Build starten).

Mit diesem Befehl wird der Build gestartet. Der Build wird asynchron ausgeführt. Die Ausgabe des Befehls ist eine JSON-Datei, die das Attribut `id` enthält. Diese Attribut-ID ist ein Verweis auf die CodeBuild Build-ID des Builds, den Sie gerade gestartet haben. Sie können den Status des Builds in der CodeBuild Konsole einsehen. Sie können auch detaillierte Protokolle zur Build-Ausführung in der Konsole einsehen. Weitere Informationen finden Sie im AWS CodeBuild Benutzerhandbuch unter [Anzeigen detaillierter Build-Informationen](#).

Wenn die aktuelle Phase `ABGESCHLOSSEN` ist, bedeutet dies, dass Ihr Build erfolgreich abgeschlossen wurde und Ihre kompilierten Artefakte auf Amazon S3 bereit sind.

## Schritt 9: Laden Sie die Ausgabeartefakte herunter

In diesem Schritt laden Sie die Ausgabeartefakte von Amazon S3 herunter. Das Micro Focus Build Tool kann verschiedene Typen von ausführbaren Dateien erstellen. In diesem Tutorial generiert es gemeinsam genutzte Objekte.

1. Melden Sie sich bei der Amazon S3 S3-Konsole an.
2. Wählen Sie im Abschnitt Buckets `role="bold">` den Namen Ihres Ausgabe-Buckets aus, zum Beispiel `codebuild-regionId-accountId-output-bucket`
3. Wählen Sie „Herunterladen“ `role="bold">`.
4. Entpacken Sie die heruntergeladene -Datei. Navigieren Sie zum Zielordner, um die Build-Artefakte zu sehen. Dazu gehören die gemeinsam genutzten `.so` Linux-Objekte.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, um zusätzliche Kosten zu vermeiden. Führen Sie dazu die folgenden Schritte aus:

- Löschen Sie die S3-Buckets, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen eines Buckets](#) im Amazon Simple Storage Service-Benutzerhandbuch.
- Löschen Sie die IAM-Richtlinien, die Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie unter [Löschen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.
- Löschen Sie die IAM-Rolle, die Sie für dieses Tutorial erstellt haben. Weitere Informationen zum Löschen von Rollen finden Sie unter [Löschen von Rollen oder Instance-Profilen](#) im IAM-Benutzerhandbuch.



- Löschen Sie das CodeBuild Projekt, das Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie [CodeBuild im AWS CodeBuild Benutzerhandbuch unter Löschen eines Build-Projekts](#).

## Tutorial: Einrichtung einer CI/CD-Pipeline zur Verwendung mit Micro Focus Enterprise Developer

In diesem Tutorial erfahren Sie, wie Sie die BankDemo Beispielanwendung in Micro Focus Enterprise Developer importieren, bearbeiten, kompilieren und ausführen und anschließend Ihre Änderungen übernehmen, um eine CI/CD-Pipeline auszulösen.

### Inhalt

- [Voraussetzungen](#)
- [Erstellen Sie eine grundlegende Infrastruktur für die CI/CD-Pipeline](#)
- [Erstellen Sie ein AWS CodeCommit Repository und eine CI/CD-Pipeline](#)
  - [Beispiel für eine YAML-Triggerdatei config\\_git.yml](#)
- [Erstellung AppStream von Enterprise Developer 2.0](#)
- [Einrichtung und Test für Enterprise Developer](#)
  - [Klonen Sie das BankDemo CodeCommit Repository in Enterprise Developer](#)
  - [Erstellen Sie ein BankDemo Mainframe-COBOL-Projekt und erstellen Sie eine Anwendung](#)
  - [Erstellen Sie eine lokale BankDemo CICS- und Batch-Umgebung zum Testen](#)
  - [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
  - [Starten Sie das Rumba 3270-Terminal](#)
  - [Führen Sie BankDemo eine Transaktion aus](#)
  - [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)
- [Übung 1: Verbessern Sie die Kreditberechnung in der BANKDEMO-Anwendung](#)
  - [Fügen Sie der Enterprise Developer Code Analysis eine Regel für die Kreditanalyse hinzu](#)
  - [Schritt 1: Führen Sie eine Codeanalyse für die Kreditberechnung durch](#)
  - [Schritt 2: Ändern Sie die CICS-BMS-Map und das COBOL-Programm und testen Sie](#)
  - [Schritt 3: Fügen Sie die Berechnung des Gesamtbetrags im COBOL-Programm hinzu](#)
  - [Schritt 4: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus](#)
- [Übung 2: Extrahieren Sie die Kreditberechnung in der Anwendung BankDemo](#)

- [Schritt 1: Umgestaltung der Routine zur Kreditberechnung in einen COBOL-Abschnitt](#)
- [Schritt 2: Extrahieren Sie die Routine zur Kreditberechnung in ein eigenständiges COBOL-Programm](#)
- [Schritt 3: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus](#)
- [Bereinigen von -Ressourcen](#)

## Voraussetzungen

Laden Sie die folgenden Dateien herunter.

- `basic-infra.yaml`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)
- `pipeline.yaml`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)
- `m2-code-sync-function.zip`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)
- `config_git.yaml`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)
- `BANKDEMO-source.zip`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)
- `BANKDEMO-exercise.zip`
  - [Aus der Region Europa \(Frankfurt\) herunterladen.](#)
  - [Aus der Region USA Ost \(Nord-Virginia\) herunterladen.](#)

Der Zweck jeder Datei ist wie folgt:

## basic-infra.yaml

Diese AWS CloudFormation Vorlage erstellt die grundlegende Infrastruktur, die für die CI/CD-Pipeline benötigt wird: VPC, Amazon S3 S3-Buckets usw.

## pipeline.yaml

Diese AWS CloudFormation Vorlage wird von einer Lambda-Funktion verwendet, um den Pipeline-Stack zu starten. Stellen Sie sicher, dass sich diese Vorlage in einem öffentlich zugänglichen Amazon S3 S3-Bucket befindet. Fügen Sie den Link zu diesem Bucket als Standardwert für den PipelineTemplateURL Parameter in der basic-infra.yaml Vorlage hinzu.

## m2-code-sync-function.zip

Diese Lambda-Funktion erstellt das CodeCommit Repository, die darauf basierende Verzeichnisstrukturconfig\_git.yaml, und startet den Pipeline-Stack mithilfe vonpipeline.yaml. Stellen Sie sicher, dass diese ZIP-Datei in einem öffentlich zugänglichen Amazon S3 S3-Bucket in allen Ländern verfügbar ist, in AWS-Regionen denen die AWS Mainframe-Modernisierung unterstützt wird. Wir empfehlen, die Datei in einem Bucket zu speichern AWS-Region und sie in alle Buckets zu replizieren. AWS-Regionen Verwenden Sie eine Benennungskonvention für den Bucket mit einem Suffix, das den spezifischen Bucket identifiziert AWS-Region (z. B.m2-cicd-deployment-source-eu-west-1), und fügen Sie das Präfix m2-cicd-deployment-source als Standardwert für den Parameter hinzu DeploymentSourceBucket und bilden Sie den vollständigen Bucket, indem Sie die AWS CloudFormation Substitutionsfunktion verwenden und !Sub {DeploymentSourceBucket}-\${AWS::Region} gleichzeitig in der Vorlage für die basic-infra.yaml Ressource auf diesen Bucket verweisen. SourceSyncLambdaFunction

## config\_git.yml

CodeCommit Definition der Verzeichnisstruktur. Weitere Informationen finden Sie unter [Beispiel für eine YAML-Triggerdatei config\\_git.yml](#).

## BANKDEMO-source.zip.

BankDemo Quellcode und Konfigurationsdatei, die aus dem CodeCommit Repository erstellt wurden.

## BANKDEMO-exercise.zip.

BankDemo Quelle für Tutorial-Übungen, die aus dem CodeCommit Repository erstellt wurden.

## Erstellen Sie eine grundlegende Infrastruktur für die CI/CD-Pipeline

Verwenden Sie die AWS CloudFormation Vorlage `basic-infra.yaml`, um den grundlegenden Infrastrukturstack der CI/CD-Pipeline über die Konsole zu erstellen. AWS CloudFormation Dieser Stack erstellt Amazon S3 S3-Buckets, in die Sie Ihren Anwendungscode und Ihre Daten hochladen, sowie eine unterstützende AWS Lambda Funktion, um andere notwendige Ressourcen wie ein AWS CodeCommit Repository und eine AWS CodePipeline Pipeline zu erstellen.

### Note

Um diesen Stack zu starten, benötigen Sie Berechtigungen zur Verwaltung von IAM, Amazon S3, Lambda AWS CloudFormation und Nutzungsberechtigungen. AWS KMS

1. [Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole unter `https://console.aws.amazon.com/cloudformation`.](https://console.aws.amazon.com/cloudformation)
2. Erstellen Sie einen neuen Stack mit einer der folgenden Optionen:
  - Wählen Sie Stapel erstellen aus. Dies ist die einzige Option, wenn Sie über eine aktuell ausgeführten Stack verfügen.
  - Wählen Sie auf der Seite Stacks die Option Create Stack aus. Diese Option ist nur sichtbar, wenn Sie keine laufenden Stacks haben.
3. Gehen Sie auf der Seite „Vorlage angeben“ wie folgt vor:
  - Wählen Sie unter Vorlage vorbereiten die Option Vorlage ist bereit aus.
  - Wählen Sie unter Vorlage angeben die Amazon S3 S3-URL als Vorlagenquelle und geben Sie je nach Ihrer eine der folgenden URLs ein AWS-Region.
    - `https://m2-us-east-1.s3.us-east-1.amazonaws.com/cicd/mf/basic-infra.yaml`
    - `https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/basic-infra.yaml`
  - Um Ihre Einstellungen zu akzeptieren, wählen Sie Weiter.

Die Seite „Stack erstellen“ wird geöffnet.

## Specify stack details

### Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

### Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

#### Networking Configuration

Do you want to use an existing VPC in your account?

If you select 'Yes', then you must provide the VPC ID and the Subnet IDs.

Which VPC ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID should be used?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Which private subnet ID in a different AZ should be used for HA?

If you selected 'Yes' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

Enter the CIDR block that should be used for the new VPC

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

CIDR bits for creating subnets. Choose 5 for /27, 6 for /26, 7 for /25, 8 for /24 range

If you selected 'No (Create one)' for UseExistingVPC, this parameter is required. Otherwise, this value will be ignored.

#### Deployment Configuration

Name of the S3 bucket which contains the source files for this stack deployment

Don't change unless you know what you are doing.

Name of the source package file for the infrastructure Lambda function

Don't change unless you know what you are doing.

Full URL of the pipeline CloudFormation template file


Don't change unless you know what you are doing.

What name prefix to use for the new S3 buckets?

A name prefix for the S3 buckets that will be created by this stack.


Nehmen Sie die folgenden Änderungen vor:

- Geben Sie die entsprechenden Werte für den Stack-Namen und die Parameter für die Netzwerkkonfiguration an.
- Die meisten Parameter in Bereitstellungskonfigurationen sind entsprechend vorab ausgefüllt, sodass Sie sie nicht ändern müssen. Ändern Sie je nach Ihrer AWS-Region die AWS CloudFormation Pipeline-Vorlage in eine der folgenden Amazon S3 S3-URLs.
  - <https://m2-us-east-1.s3.amazonaws.com/cicd/mf/pipeline.yaml>
  - <https://m2-eu-central-1.s3.eu-central-1.amazonaws.com/cicd/mf/pipeline.yaml>
- Wählen Sie Weiter aus.

 Note

Ändern Sie die Standardparameterwerte nur, wenn Sie die AWS CloudFormation Vorlage selbst geändert haben.

4. Wählen Sie unter Stack-Optionen konfigurieren die Option Weiter aus.
5. Wählen Sie unter Funktionen die Option Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen erstellt werden, um die Erlaubnis AWS CloudFormation zu erteilen, in Ihrem Namen eine IAM-Rolle zu erstellen. Wählen Sie Stack erstellen aus.

 Note

Es kann 3 bis 5 Minuten dauern, bis dieser Stack bereitgestellt ist.

6. Nachdem der Stack erfolgreich erstellt wurde, navigieren Sie zum Abschnitt Outputs des neu bereitgestellten Stacks. Dort finden Sie den Amazon S3 S3-Bucket, in den Sie Ihren Mainframe-Code und die abhängigen Dateien hochladen müssen.

Stack info	Events	Resources	Outputs	Parameters	Template	Change sets
<b>Outputs (7)</b>						
<input type="text" value="Search outputs"/>						
Key	Value	Description				
M2CICDNewPrivateSubnet1	subnet-0e1dda3ae86f025da	Subnet 1 for M2 CI/CD				
M2CICDNewPrivateSubnet2	subnet-0b89e607975284f8f	Subnet 2 for M2 CI/CD				
M2CICDNewVPC	vpc-034cbfc880b73dd28	VPC Id for M2 CI/CD				
MainframeCodeBucketS3URI	s3://mf-code-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Code S3 Bucket				
MainframeCodeBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-code-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Code S3 Bucket				
MainframeDataBucketS3URI	s3://mf-data-685ccc90-804004798367-us-east-1/	S3 URI to the Mainframe Test Data S3 Bucket				
MainframeDataBucketURL	<a href="https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects">https://s3.console.aws.amazon.com/s3/buckets/mf-data-685ccc90-804004798367-us-east-1?region=us-east-1&amp;tab=objects</a>	Management Console URL to the Mainframe Test Data S3 Bucket				

## Erstellen Sie ein AWS CodeCommit Repository und eine CI/CD-Pipeline

In diesem Schritt erstellen Sie ein CodeCommit Repository und stellen einen CI/CD-Pipeline-Stack bereit, indem Sie eine Lambda-Funktion aufrufen, die AWS CloudFormation zur Erstellung des Pipeline-Stacks aufruft.

1. Laden Sie die [BankDemo Beispielanwendung](#) auf Ihren lokalen Computer herunter.
2. Laden Sie `bankdemo.zip` von Ihrem lokalen Computer in den Amazon S3 S3-Bucket hoch, der in erstellt wurde [Erstellen Sie eine grundlegende Infrastruktur für die CI/CD-Pipeline](#).
3. Laden Sie `config_git.yml` herunter.
4. Ändern Sie das bei `config_git.yml` Bedarf wie folgt:

- Fügen Sie Ihren eigenen Ziel-Repository-Namen, Ziel-Branch und Commit-Nachricht hinzu.

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
```

- Fügen Sie die E-Mail-Adresse hinzu, an die Sie Benachrichtigungen erhalten möchten.

```

pipeline-config:
  # Send pipeline failure notifications to these email addresses
  alert-notifications:
    - myname@mycompany.com
  # Send notifications for manual approval before production deployment to these
  email addresses
  approval-notifications:
    - myname@mycompany.com

```

- Laden Sie die `config_git.yml` Datei mit der Definition der CodeCommit Repository-Ordnerstruktur in den Amazon S3 S3-Bucket hoch, der in erstellt wurde [Erstellen Sie eine grundlegende Infrastruktur für die CI/CD-Pipeline](#). Dadurch wird die Lambda-Funktion aufgerufen, die das Repository und die Pipeline automatisch bereitstellt.

Dadurch wird ein CodeCommit Repository mit dem Namen erstellt, der in der in der `config_git.yml` Datei `target-repository` definierten Namen angegeben ist; zum Beispiel `bankdemo-repo`

Die Lambda-Funktion erstellt auch den CI/CD-Pipeline-Stack durch. AWS CloudFormation Der AWS CloudFormation Stack wird dasselbe Präfix wie der angegebene `target-repository` Name haben, gefolgt von einer zufälligen Zeichenfolge (z. B. `bankdemo-repo-01234567`). Sie finden die CodeCommit Repository-URL und die URL für den Zugriff auf die erstellte Pipeline in der AWS Management Console.

**bankdemo-repo-mcdilnof** Delete Update Stack actions ▼ Create stack ▼

Stack info | Events | Resources | **Outputs** | Parameters | Template | Change sets

**Outputs (2)** ↻

⚙️

Key	Value	Description
CodeCommitRepo	<a href="https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo">https://git-codecommit.us-west-2.amazonaws.com/v1/repos/bankdemo-repo</a>	HTTPS endpoint to clone the CodeCommit repository
PipelineURL	<a href="https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2">https://us-west-2.console.aws.amazon.com/codesuite/codepipeline/pipelines/bankdemo-repo-mcdilnof-M2Pipeline-17WYBNGCXB82K/view?region=us-west-2</a>	URL to access the pipeline on AWS Management Console

- Wenn die Erstellung des CodeCommit Repositorys abgeschlossen ist, wird die CI/CD-Pipeline sofort ausgelöst, um eine vollständige CI/CD durchzuführen.



7. Sobald die Datei per Push übertragen wurde, löst sie automatisch die Pipeline aus. Diese wird erstellt, im Staging bereitgestellt, einige Tests ausgeführt und auf die manuelle Genehmigung gewartet, bevor sie in der Produktionsumgebung bereitgestellt wird.

## Beispiel für eine YAML-Triggerdatei config\_git.yml

```
repository-config:
  target-repository: bankdemo-repo
  target-branch: main
  commit-message: Initial commit for bankdemo-repo main branch
  directory-structure:
    - '/':
      files:
        - build.xml
        - '*.yaml'
        - '*.yml'
        - '*.xml'
        - 'LICENSE.txt'
      readme: |
        # Root Folder
        - 'build.xml' : Build configuration for the application
    - tests:
      files:
        - '*.py'
      readme: |
        # Test Folder
        - '*.py' : Test scripts
    - config:
      files:
        - 'BANKDEMO.csd'
        - 'BANKDEMO.json'
        - 'BANKDEMO_ED.json'
        - 'dfhdrdat'
        - 'ESPGSQLXA.dll'
        - 'ESPGSQLXA64.so'
        - 'ESPGSQLXA64_S.so'
        - 'EXTFH.cfg'
        - 'm2-2021-04-28.normal.json'
        - 'MFDBFH.cfg'
        - 'application-definition-template-config.json'
      readme: |
        # Config Folder
```

This folder contains the application configuration files.

- 'BANKDEMO.csd' : CICS Resource definitions export file
- 'BANKDEMO.json' : Enterprise Server configuration
- 'BANKDEMO\_ED.json' : Enterprise Server configuration for ED
- 'dfhdrdat' : CICS resource definition file
- 'ESPGSQLXA.dll' : XA switch module Windows
- 'ESPGSQLXA64.so' : XA switch module Linux
- 'ESPGSQLXA64\_S.so' : XA switch module Linux
- 'EXTFH.cfg' : Micro Focus File Handler configuration
- 'm2-2021-04-28.normal.json' : M2 request document
- 'MFDBFH.cfg' : Micro Focus Database File Handler
- 'application-definition-template-config.json' : Application definition for

M2

- source:
  - subdirs:
    - .settings:
      - files:
        - '.bms.mfdirset'
        - '.cbl.mfdirset'
    - copybook:
      - files:
        - '\*.cpy'
        - '\*.inc'
      - readme: |
        - # Copy folder
        - This folder contains the source for COBOL copy books, PLI includes, ...
        - .cpy COBOL copybooks
        - .inc PLI includes
  - # - ctlcards:
    - # files:
      - # - '\*.ctl'
      - # - 'KBNKSRT1.txt'
    - # readme: |
      - # # Control Card folder
      - # This folder contains the source for Batch Control Cards
      - # - .ctl Control Cards
  - ims:
    - files:
      - '\*.dbd'
      - '\*.psb'
    - readme: |
      - # ims folder
      - This folder contains the IMS DB source files with the extensions
      - .dbd for IMS DBD source

```
    - .psb for IMS PSB source
- jcl:
  files:
    - '*.jcl'
    - '*.ctl'
    - 'KBNKSRT1.txt'
    - '*.prc'
  readme: |
    # jcl folder
    This folder contains the JCL source files with the extensions
    - .jcl
#
# - proclib:
#   files:
#     - '*.prc'
#   readme: |
#     # proclib folder
#     This folder contains the JCL procedures referenced via PROCLIB
#     statements in the JCL with extensions
#     - .prc
- rdbms:
  files:
    - '*.sql'
  readme: |
    # rdbms folder
    This folder contains any DB2 related source files with extensions
    - .sql for any kind of SQL source
- screens:
  files:
    - '*.bms'
    - '*.mfs'
  readme: |
    # screens folder
    This folder contains the screens source files with the extensions
    - .bms for CICS BMS screens
    - .mfs for IMS MFS screens
  subdirs:
    - .settings:
      files:
        - '*.bms.mfdirset'
- cobol:
  files:
    - '*.cbl'
    - '*.pli'
  readme: |
```

```
    # source folder
    This folder contains the program source files with the extensions
    - .cbl for COBOL source
    - .pli for PLI source
  subdirs:
  - .settings:
      files:
        - '*.cbl.mfdirset'
- tests:
  files:
  - 'test_script.py'
  readme: |
    # tests Folder
    This folder contains the application test scripts
pipeline-config:
  alert-notifications:
  - myname@mycompany.com
  approval-notifications:
  - myname@mycompany.com
```

## Erstellung AppStream von Enterprise Developer 2.0

Informationen zur Einrichtung von Micro Focus Enterprise Developer auf AppStream 2.0 finden Sie unter [Tutorial: Micro Focus Enterprise Developer auf AppStream 2.0 einrichten](#).

Um das CodeCommit Repository mit Enterprise Developer zu verbinden, verwenden Sie den `target-repository` in angegebenen Namen [Beispiel für eine YAML-Triggerdatei config\\_git.yml](#).


## Einrichtung und Test für Enterprise Developer

### Themen

- [Klonen Sie das BankDemo CodeCommit Repository in Enterprise Developer](#)
- [Erstellen Sie ein BankDemo Mainframe-COBOL-Projekt und erstellen Sie eine Anwendung](#)
- [Erstellen Sie eine lokale BankDemo CICS- und Batch-Umgebung zum Testen](#)
- [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
- [Starten Sie das Rumba 3270-Terminal](#)
- [Führen Sie BankDemo eine Transaktion aus](#)
- [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)

Connect zu der Enterprise Developer AppStream 2.0-Instanz her, in der Sie sie erstellt haben [Erstellung AppStream von Enterprise Developer 2.0](#).

1. Starten Sie Enterprise Developer von Windows Start aus. Wählen Sie Micro Focus Enterprise Developer und anschließend Enterprise Developer for Eclipse. Wenn Sie zum ersten Mal starten, kann es einige Zeit dauern.
2. Geben Sie im Eclipse Launcher in Workspace: Geben Sie die `C:\Users\\workspace` Eingabetaste ein und wählen Sie Launch.


 Note

Stellen Sie sicher, dass Sie denselben Standort wählen, nachdem Sie erneut eine Verbindung zur AppStream 2.0-Instanz hergestellt haben. Die Workspace-Auswahl ist nicht dauerhaft.

3. Wählen Sie unter Willkommen die Option Open COBOL Perspective aus. Dies wird nur beim ersten Mal für einen neuen Workspace angezeigt.

## Klonen Sie das BankDemo CodeCommit Repository in Enterprise Developer

1. Wählen Sie Fenster/Perspektive/Perspektive öffnen/Andere... /Git.
2. Wähle Ein Git-Repository klonen.
3. Geben Sie in Clone Git Repository die folgenden Informationen ein:
  - Geben Sie im Feld Standort-URI die HTTPS-URL des CodeCommit Repositorys ein.

 Note

Kopieren Sie die Klon-URL HTTPS für das CodeCommit Repository in die AWS Management Console und fügen Sie sie hier ein. Die URI wird in die Host - und Repository-Pfade aufgeteilt.

- Geben Sie die CodeCommit Benutzerrepository-Anmeldeinformationen unter Authentifizierungsbenutzer und Passwort ein und wählen Sie In Secure Store speichern aus.
4. Wählen Sie unter Branch Selection die Option Main branch und dann Next aus.
  5. Geben Sie im Feld Lokales Ziel im Feld Verzeichnis den Text Fertig stellen ein `C:\Users\\workspace` und wählen Sie ihn aus.

Der Klonvorgang ist abgeschlossen, wenn dies `BANKDEMO [main]` in der Git-Repository-Ansicht angezeigt wird.

## Erstellen Sie ein BankDemo Mainframe-COBOL-Projekt und erstellen Sie eine Anwendung

1. Wechseln Sie zur COBOL-Perspektive.
2. Deaktivieren Sie in Project die Option Automatisch erstellen.
3. Wählen Sie unter Datei die Option Neu und dann Mainframe-COBOL-Projekt aus.
4. Geben Sie im Feld Neues Mainframe-COBOL-Projekt die folgenden Informationen ein:
  - Geben Sie im Feld Projektname den Wert ein. BankDemo
  - Wählen Sie die Micro Focus-Vorlage [64 Bit].
  - Wählen Sie Finish (Abschließen).
5. Erweitern Sie im COBOL Explorer das neue BankDemo Projekt.

### Note

[`BANKDEMO main`] bedeutet in eckigen Klammern, dass das Projekt mit dem lokalen BankDemo CodeCommit Repository verbunden ist.

6. Wenn in der Strukturansicht keine Einträge für COBOL-Programme, Copybooks, BMS-Quelle und JCL-Dateien angezeigt werden, wählen Sie im Kontextmenü des Projekts die BankDemo Option Aktualisieren.
7. Wählen Sie im BankDemo Kontextmenü Eigenschaften/Micro Focus/Projekteinstellungen/COBOL:
  - Wählen Sie Zeichensatz — ASCII.
  - Wählen Sie Anwenden und dann Schließen.
8. Wenn der Build der BMS- und COBOL-Quelle nicht sofort startet, überprüfen Sie im Projektmenü, ob die Option Automatisch erstellen aktiviert ist.

Die Build-Ausgabe wird in der Konsolenansicht angezeigt und sollte nach einigen Minuten mit den Meldungen `BUILD SUCCESSFUL` und abgeschlossen sein. `Build finished with no errors`

Die BankDemo Anwendung sollte jetzt kompiliert und für die lokale Ausführung bereit sein.

## Erstellen Sie eine lokale BankDemo CICS- und Batch-Umgebung zum Testen

1. Erweitern Sie im COBOL Explorer. `BANKDEMO / config`
2. Öffnen `BANKDEMO_ED.json` Sie im Editor.
3. Suchen Sie die Zeichenfolge `ED_Home=` und ändern Sie den Pfad so, dass er auf das Enterprise Developer-Projekt verweist, wie folgt:`D:\\<username>\\workspace\\BANKDEMO`. Beachten Sie die Verwendung von doppelten Schrägstrichen (\\) in der Pfaddefinition.
4. Speichern und schließen Sie die Datei.
5. Wählen Sie Server Explorer.
6. Wählen Sie im Standard-Kontextmenü die Option Administrationsseite öffnen. Die Micro Focus Enterprise Server-Administrationsseite wird im Standardbrowser geöffnet.
7. Nehmen Sie nur für AppStream 2.0-Sitzungen die folgenden Änderungen vor, damit Sie Ihre lokale Enterprise Server-Region für lokale Tests beibehalten können:
  - Wählen Sie unter Directory Server//Default die Option PROPERTIES//Configuration aus.
  - Ersetzen Sie den Repository-Speicherort durch `D:\\<username>\\My Files\\Home Folder\\MFDS`.

### Note

Sie müssen die Schritte 5 bis 8 nach jeder neuen Verbindung zu einer AppStream 2.0-Instanz ausführen.

8. Wählen Sie unter Directory Server//Default die Option Import aus und führen Sie dann die folgenden Schritte aus:
  - Wählen Sie in Schritt 1: Importtyp die Option JSON und dann Weiter aus.
  - Klicken Sie in Schritt 2: Hochladen auf die Datei im blauen Quadrat, um sie hochzuladen.
  - Geben Sie unter Datei zum Hochladen auswählen Folgendes ein:
    - Name der Datei:`D:\\<username>\\workspace\\BANKDEMO\\config\\BANKDEMO_ED.json`.
    - Klicken Sie auf Open.

- Wählen Sie Weiter.
- Löschen Sie in Schritt 3: Regionen die Option Ports von Endpunkten löschen.
- Wählen Sie Weiter aus.
- Wählen Sie in Schritt 4: Import die Option Import aus.
- Wählen Sie Finish (Abschließen).

In der Liste wird nun ein neuer Servername angezeigt BANKDEMO.

## Starten Sie den BANKDEMO Server von Enterprise Developer aus

1. Wählen Sie Enterprise Developer.
2. Wählen Sie im Server Explorer die Option Standard und anschließend im Kontextmenü die Option Aktualisieren aus.

In der Serverliste sollte jetzt auch BANKDEMO angezeigt werden.

3. Wählen Sie BANKDEMO.
4. Wählen Sie im Kontextmenü die Option Mit Projekt verknüpfen und anschließend BANKDEMO.
5. Wählen Sie im Kontextmenü die Option Start.

In der Konsolenansicht sollte das Protokoll für den Serverstart angezeigt werden.

Wenn die Meldung angezeigt BANKDEMO CASSI5030I PLTPI Phase 2 List(PI) Processing Completed wird, ist der Server bereit, die CICS BANKDEMO-Anwendung zu testen.

## Starten Sie das Rumba 3270-Terminal

1. Starten Sie von Windows Start aus Micro Focus Rumba+ Desktop/Rumba+ Desktop.
2. Wählen Sie unter Willkommen die Option NEUE SITZUNG ERSTELLEN/Mainframe Display aus.
3. Wählen Sie unter Mainframe Display die Option Connection/Configure aus.
4. Wählen Sie in der Sitzungskonfiguration die Option Connection/TN3270 aus.
5. Wählen Sie unter Hostname/Address die Option Einfügen und geben Sie die IP-Adresse ein.  
127.0.0.1
6. Geben Sie im Feld Telnet-Port den Wert port 6000 ein.



7. Wählen Sie Apply (Anwenden) aus.
8. Wählen Sie Connect aus.

Auf dem CICS-Willkommensbildschirm wird ein Bildschirm mit der Meldung in Zeile 1 angezeigt:  
This is the Micro Focus MFE CICS region BANKDEMO

9. Drücken Sie Strg+Shift+Z, um den Bildschirm zu löschen.

## Führen Sie BankDemo eine Transaktion aus

1. Geben Sie in einem leeren Bildschirm einBANK.
2. Auf dem Bildschirm BANK10, im Eingabefeld für die Benutzer-ID... :, geben Sie ein guest und drücken Sie die Eingabetaste.
3. Geben Sie auf dem Bildschirm BANK20 in das Eingabefeld vor Berechne die Kreditkosten ein / (Schrägstrich) ein und drücken Sie die Eingabetaste.
4. Auf dem Bildschirm BANK70:
  - In dem Betrag, den Sie leihen möchten... :, geben Sie ein10000.
  - In Bei einem Zinssatz von... :, geben Sie ein5.0.
  - In Für wie viele Monate... :, geben Sie ein10.
  - Drücken Sie die Eingabetaste.

Das folgende Ergebnis sollte angezeigt werden:

```
Resulting monthly payment.....:  $1023.06
```

Damit ist die Einrichtung der BANKDEMO-Anwendung in Enterprise Developer abgeschlossen.

## Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus

1. Wählen Sie im Server-Explorer die Option Standard und anschließend im Kontextmenü die Option Aktualisieren aus.
2. Wählen Sie BANKDEMO.
3. Wählen Sie im Kontextmenü die Option Stopp.

In der Konsolenansicht sollte das Protokoll für den Serverstopp angezeigt werden.

Wenn die Meldung angezeigt `Server: BANKDEMO stopped successfully` wird, wurde der Server erfolgreich heruntergefahren.

## Übung 1: Verbessern Sie die Kreditberechnung in der BANKDEMO-Anwendung

### Themen

- [Fügen Sie der Enterprise Developer Code Analysis eine Regel für die Kreditanalyse hinzu](#)
- [Schritt 1: Führen Sie eine Codeanalyse für die Kreditberechnung durch](#)
- [Schritt 2: Ändern Sie die CICS-BMS-Map und das COBOL-Programm und testen Sie](#)
- [Schritt 3: Fügen Sie die Berechnung des Gesamtbetrags im COBOL-Programm hinzu](#)
- [Schritt 4: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus](#)

In diesem Szenario führen Sie den Prozess durch, bei dem Sie eine Beispieländerung am Code vornehmen, ihn bereitstellen und testen.

Die Kreditabteilung möchte auf dem Bildschirm BANK70 für die Kreditberechnung ein neues Feld einrichten, in dem der gesamte Kreditbetrag angezeigt wird. Dazu muss der BMS-Bildschirm MBANK70.CBL geändert und ein neues Feld und das entsprechende Bildschirmverwaltungsprogramm SBANK70P.CBL mit den zugehörigen Copybooks hinzugefügt werden. Darüber hinaus muss die Kreditberechnungsroutine in BBANK70P.CBL um die zusätzliche Formel erweitert werden.

Stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen, um diese Übung abzuschließen.

- Laden Sie [BANKDEMO-exercise.zip](#) herunter auf `D:\PhotonUser\My Files\Home Folder`.
- Extrahieren Sie die Zip-Datei nach `D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise`.
- Ordner erstellen `D:\PhotonUser\My Files\Home Folder\AnalysisRules`.
- Kopieren Sie die Regeldatei `Loan+Calculation+Update.General-1.xml` aus dem `BANKDEMO-exercise` Ordner in `D:\PhotonUser\My Files\Home Folder\AnalysisRules`.

**Note**

Codeänderungen in \*.CBL und \*.CPY sind für diese Übung in den Spalten 1—6 mit EXER01 gekennzeichnet.

Fügen Sie der Enterprise Developer Code Analysis eine Regel für die Kreditanalyse hinzu

In Micro Focus Enterprise Analyzer definierte Analyseregeln können aus Enterprise Analyzer exportiert und in Enterprise Developer importiert werden, um dieselben Analyseregeln für alle Quellen im Enterprise Developer-Projekt auszuführen.

1. Öffnen Sie `Window/Preferences/Micro Focus/COBOL/Code Analysis/Rules`.
2. Wählen Sie `Bearbeiten...` und geben Sie den Namen des Ordners ein `D:\PhotonUser\My Files\Home Folder\AnalysisRules`, der die Regeldatei enthält `Loan+Calculation+Update.General-1.xml`.
3. Wählen Sie `Finish` (Abschließen).
4. Wählen Sie „Anwenden“ und anschließend „Schließen“.
5. Wählen Sie im Kontextmenü des BANKDEMO-Projekts die Option `Codeanalyse` aus.

Sie sollten einen Eintrag für die Aktualisierung der Kreditberechnung sehen.

**Schritt 1: Führen Sie eine Codeanalyse für die Kreditberechnung durch**

Mit der neuen Analyseregeln wollen wir die COBOL-Programme und Codezeilen identifizieren, die den Suchmustern entsprechen `*PAYMENT*`, `*LOAN*` und zwar `*RATE*` in Ausdrücken, Anweisungen und Variablen. Dies hilft dabei, durch den Code zu navigieren und erforderliche Codeänderungen zu identifizieren.

1. Wählen Sie im KONTEXTMENÜ des BANKDEMO-Projekts die Option `Codeanalyse/Aktualisierung der Kreditberechnung`.

Dadurch wird die Suchregel ausgeführt und die Ergebnisse werden auf einer neuen Registerkarte namens `Codeanalyse` aufgelistet. Der Analyselauf ist abgeschlossen, wenn der grüne Fortschrittsbalken unten rechts verschwindet.

Auf der Registerkarte Codeanalyse sollte eine erweiterte Liste BBANK20P.CBL mit Anweisungen SBANK70P.CBL, Ausdrücken BBANK70P.CBL und Variablen angezeigt werden, die den Suchmustern entsprechen.

Im Ergebnis werden nur Literale verschoben, die mit dem Suchmuster übereinstimmen. BBANK20P.CBL Dieses Programm kann also ignoriert werden.

2. Wählen Sie in der Tab-Menüleiste - Symbol, um alles zu reduzieren.
3. Erweitern Sie SBANK70P.CBL und wählen Sie mit einem Doppelklick alle Zeilen in beliebiger Reihenfolge aus, um zu sehen, wie dadurch die Quelle geöffnet und die im Quellcode ausgewählte Zeile hervorgehoben wird. Sie werden auch erkennen, dass alle identifizierten Quellzeilen markiert sind.

## Schritt 2: Ändern Sie die CICS-BMS-Map und das COBOL-Programm und testen Sie

Zuerst werden wir die BMS-Map MBANK70.BMS sowie das Bildschirmverwaltungsprogramm SBANK70P.CBL und das Copybook CBANKDAT.CPY so ändern, dass das neue Feld angezeigt wird. Um unnötiges Programmieren in dieser Übung zu vermeiden, sind modifizierte Quellmodule im D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01 Ordner verfügbar. Normalerweise würde ein Entwickler die Ergebnisse der Codeanalyse verwenden, um in den Quellen zu navigieren und sie zu ändern. Wenn Sie Zeit haben und die manuellen Änderungen vornehmen möchten, tun Sie dies anhand der Informationen unter \*Manuelle Änderung in MBANK70.BMS und SBANK70P.CBL (optional) \*.

Kopieren Sie für schnelle Änderungen die folgenden Dateien:

1. ..\BANKDEMO-exercise\Exercise01\screens\MBANK70.BMS auf D:\PhotonUser\workspace\bankdemo\source\screens.
2. ..\BANKDEMO-exercise\Exercise01\cobol\SBANK70P.CBL auf D:\PhotonUser\workspace\bankdemo\source\cobol.
3. ..\BANKDEMO-exercise\Exercise01\copybook\CBANKDAT.CPY auf D:\PhotonUser\workspace\bankdemo\source\copybook.
4. Um sicherzustellen, dass alle von den Änderungen betroffenen Programme kompiliert sind, wählen Sie Project/Clean... /Gesamtes Projekt säubern.

Gehen Sie wie folgt vor `SBANK70P.CBL`, um manuelle Änderungen an `MBANK70.BMS` und vorzunehmen:

- Für manuelle Änderungen in der `MBANK70.BMS` BMS-Quelle fügen Sie nach dem `PAYMENT` Feld Folgendes hinzu:
  - `TXT09` mit denselben Attributen wie `TXT08` und dem `ANFANGSWert` „Gesamtkreditbetrag“
  - `INSGESAMT` mit denselben Attributen wie `PAYMENT`

### Änderungen testen

Um die Änderungen zu testen, wiederholen Sie die Schritte in den folgenden Abschnitten:

1. [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
2. [Starten Sie das Rumba 3270-Terminal](#)
3. [Führen Sie BankDemo eine Transaktion aus](#)

Außerdem solltest du jetzt auch den Text sehen `Total Loan Amount` . . . . . :

4. [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)

### Schritt 3: Fügen Sie die Berechnung des Gesamtbetrags im COBOL-Programm hinzu

Im zweiten Schritt werden wir die Berechnung für den gesamten Kreditbetrag ändern `BBANK70P.CBL` und hinzufügen. Die vorbereitete Quelle mit den erforderlichen Änderungen ist im `D:\PhotonUser\My Files\Home Folder\BANKDEMO-exercise\Exercise01` Ordner verfügbar. Wenn Sie Zeit haben und die manuellen Änderungen vornehmen möchten, tun Sie dies anhand der Informationen unter *\*Manuelle Änderung in `BBANK70P.CBL` (optional) \**.

Kopieren Sie zur schnellen Änderung die folgende Datei:

- `..\BANKDEMO-exercise\Exercise01\source\cobol\BBANK70P.CBL` auf `D:\PhotonUser\workspace\bankdemo\source\cobol`.

Gehen Sie wie folgt vor `BBANK70P.CBL`, um eine manuelle Änderung an vorzunehmen:

- Identifizieren Sie anhand des Ergebnisses der Codeanalyse die erforderlichen Änderungen.

## Testen Sie die Änderungen

Um die Änderungen zu testen, wiederholen Sie die Schritte in den folgenden Abschnitten:

1. [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
2. [Starten Sie das Rumba 3270-Terminal](#)
3. [Führen Sie BankDemo eine Transaktion aus](#)

Außerdem solltest du jetzt auch den Text sehen  
Total Loan Amount.....: \$10230.60.

4. [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)

## Schritt 4: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus

Übernehmen Sie die Änderungen im zentralen CodeCommit Repository und lösen Sie die CI/CD-Pipeline aus, um die Änderungen zu erstellen, zu testen und bereitzustellen.

1. Wählen Sie im BANKDEMO-Projekt im Kontextmenü die Option Team/Commit aus.
2. Geben Sie auf der Registerkarte Git Staging die folgende Commit-Nachricht ein: Added Total Amount Calculation.
3. Wählen Sie Commit und Push... .
4. Öffnen Sie die CodePipeline Konsole und überprüfen Sie den Status der Pipeline-Ausführung.

### Note

Falls Sie Probleme mit der Enterprise Developer- oder Teams-Funktion Commit oder Push haben, verwenden Sie die Git Bash-Befehlszeilenschnittstelle.

## Übung 2: Extrahieren Sie die Kreditberechnung in der Anwendung BankDemo

### Themen

- [Schritt 1: Umgestaltung der Routine zur Kreditberechnung in einen COBOL-Abschnitt](#)
- [Schritt 2: Extrahieren Sie die Routine zur Kreditberechnung in ein eigenständiges COBOL-Programm](#)

- [Schritt 3: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus](#)

In der nächsten Übung bearbeiten Sie ein weiteres Beispiel für einen Änderungsantrag. In diesem Szenario möchte die Kreditabteilung die Kreditberechnungsroutine als eigenständige Routine wiederverwenden WebService. Die Routine sollte in COBOL verbleiben und auch weiterhin vom vorhandenen CICS COBOL-Programm aus aufgerufen werden können. BBANK70P.CBL

## Schritt 1: Umgestaltung der Routine zur Kreditberechnung in einen COBOL-Abschnitt

Im ersten Schritt extrahieren wir die Routine zur Kreditberechnung in einen COBOL-Abschnitt. Dieser Schritt ist erforderlich, um den Code im nächsten Schritt in ein eigenständiges COBOL-Programm zu extrahieren.

1. BBANK70P.CBL im COBOL-Editor öffnen.
2. Wählen Sie im Editor aus dem Kontextmenü Code Analysis/Loan Calculation Update. Dadurch wird nur die aktuelle Quelle nach Mustern durchsucht, die in der Analyseregel definiert sind.
3. Suchen Sie im Ergebnis auf der Registerkarte Codeanalyse nach der ersten arithmetischen Anweisung. DIVIDE WS-LOAN-INTEREST BY 12
4. Doppelklicken Sie auf die Anweisung, um zur Quellzeile im Editor zu navigieren. Dies ist die erste Aussage der Kreditberechnungsroutine.
5. Markieren Sie den folgenden Codeblock für die Kreditberechnungsroutine, die in einen Abschnitt extrahiert werden soll.

```

DIVIDE WS-LOAN-INTEREST BY 12
      GIVING WS-LOAN-INTEREST ROUNDED.
COMPUTE WS-LOAN-MONTHLY-PAYMENT ROUNDED =
      ((WS-LOAN-INTEREST * ((1 + WS-LOAN-INTEREST)
      ** WS-LOAN-TERM)) /
      (((1 + WS-LOAN-INTEREST) * WS-LOAN-TERM) - 1 ))
      * WS-LOAN-PRINCIPAL.
EXER01 COMPUTE WS-LOAN-TOTAL-PAYMENT =
EXER01      (WS-LOAN-MONTHLY-PAYMENT * WS-LOAN-TERM).

```

6. Wählen Sie im Kontextmenü des Editors die Option Refactor/In Abschnitt extrahieren... .
7. Geben Sie den Namen des neuen Abschnitts ein: LOAN-CALCULATION.
8. Wählen Sie OK aus.

Der markierte Codeblock wurde nun in den neuen LOAN-CALCULATION Abschnitt extrahiert und der Codeblock wurde durch die PERFROM LOAN-CALCULATION Anweisung ersetzt.

## Änderungen testen

Um die Änderungen zu testen, wiederholen Sie die in den folgenden Abschnitten beschriebenen Schritte.

1. [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
2. [Starten Sie das Rumba 3270-Terminal](#)
3. [Führen Sie BankDemo eine Transaktion aus](#)

Außerdem solltest du jetzt auch den Text sehen  
Total Loan Amount.....: \$10230.60.

4. [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)

### Note

Wenn Sie die obigen Schritte zum Extrahieren des Codeblocks in einen Abschnitt vermeiden möchten, können Sie den modifizierten Quellcode für Schritt 1 von ..  
\BANKDEMO-exercise\Exercis02\Step1\cobl\BBANK70P.CBL bis kopierend:  
\PhotonUser\workspace\bankdemo\source\cobl.

## Schritt 2: Extrahieren Sie die Routine zur Kreditberechnung in ein eigenständiges COBOL-Programm

In Schritt 2 wird der Codeblock in LOAN-CALCULATION diesem Abschnitt in ein eigenständiges Programm extrahiert und der ursprüngliche Code wird durch Code zum Aufrufen des neuen Unterprogramms ersetzt.

1. Öffnen Sie BBANK70P.CBL den Editor und suchen Sie nach der neuen PERFORM LOAN-CALCULATION Anweisung, die Sie in Schritt 1 erstellt haben.
2. Platzieren Sie den Cursor innerhalb des Abschnittsnamens. Er wird grau markiert.
3. Wählen Sie im Kontextmenü Refactor->Abschnitt/Absatz in Programm extrahieren... .



4. Geben Sie im Feld Abschnitt/Absatz in Programm extrahieren den Namen Neuer Dateiname ein: LOANCALC.CBL.
5. Wählen Sie OK aus.

Das neue Programm wird im Editor geöffnet LOANCALC . CBL.

6. Scrollen Sie nach unten und überprüfen Sie den Code, der für die Anrufoberfläche extrahiert und generiert wird.
7. Wählen Sie den Editor mit BBANK70P . CBL und gehen Sie zu LOAN-CALCULATION SECTION. Überprüfen Sie den Code, der generiert wird, um das neue Unterprogramm LOANCALC . CBL aufzurufen.

#### Note

Die CALL Anweisung verwendet DFHEIBLK und DFHCOMMAREA ruft LOANCALC mit CICS-Steuerblöcken auf. Da wir das neue LOANCALC . CBL Unterprogramm als Nicht-CICS-Programm aufrufen wollen, müssen wir DFHEIBLK und DFHCOMMAREA aus dem Aufruf entfernen, entweder durch Auskommentieren oder Löschen.

## Änderungen testen

Um die Änderungen zu testen, wiederholen Sie die in den folgenden Abschnitten beschriebenen Schritte.

1. [Starten Sie den BANKDEMO Server von Enterprise Developer aus](#)
2. [Starten Sie das Rumba 3270-Terminal](#)
3. [Führen Sie BankDemo eine Transaktion aus](#)

Außerdem solltest du jetzt auch den Text sehen  
Total Loan Amount.....: \$10230.60.

4. [Stoppen Sie den BANKDEMO-Server von Enterprise Developer aus](#)

#### Note

Wenn Sie die obigen Schritte zum Extrahieren des Codeblocks in einen Abschnitt vermeiden möchten, können Sie den modifizierten Quellcode für Schritt 1 von ..\BANKDEMO-exercise\Exercis02\Step2\cobo1\BBANK70P.CBL und

```
LOANCALC.CBL nach kopierenD:\PhotonUser\workspace\bankdemo\source  
\cobol.
```

### Schritt 3: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus

Übernehmen Sie die Änderungen im zentralen CodeCommit Repository und lösen Sie die CI/CD-Pipeline aus, um die Änderungen zu erstellen, zu testen und bereitzustellen.

1. Wählen Sie im BANKDEMO-Projekt im Kontextmenü die Option Team/Commit.
2. Auf der Registerkarte Git Staging
  - Fügen Sie in den Unstaged Stages LOANCALC.CBL und LoanCalc.CBL.MFDirSet hinzu.
  - Geben Sie eine Commit-Nachricht ein `Added Total Amount Calculation:`.
3. Wählen Sie Commit und Push... .
4. Öffnen Sie die CodePipeline Konsole und überprüfen Sie den Status der Pipeline-Ausführung.

#### Note

Falls Sie Probleme mit der Enterprise Developer- oder Teams-Funktion Commit oder Push haben, verwenden Sie die Git Bash-Befehlszeilenschnittstelle.

## Bereinigen von -Ressourcen

Wenn Sie die Ressourcen, die Sie für dieses Tutorial erstellt haben, nicht mehr benötigen, löschen Sie sie, damit Ihnen keine weiteren Gebühren berechnet werden. Führen Sie folgende Schritte aus:

- Löschen Sie die CodePipeline Pipeline. Weitere Informationen finden Sie unter [Löschen einer Pipeline CodePipeline im AWS CodePipeline Benutzerhandbuch](#).
- Löschen Sie das CodeCommit Repository. Weitere Informationen finden Sie im AWS CodeCommit Benutzerhandbuch unter [Löschen eines CodeCommit Repositorys](#).
- Löschen Sie den S3; -Bucket. Weitere Informationen finden Sie unter [Löschen eines Buckets im Amazon Simple Storage Service-Benutzerhandbuch](#).

- Löschen Sie den AWS CloudFormation Stapel. Weitere Informationen finden Sie im AWS CloudFormation Benutzerhandbuch unter [Löschen eines Stacks auf der AWS CloudFormation Konsole](#).

## Batch-Dienstprogramme bei der AWS Mainframe-Modernisierung

Mainframe-Anwendungen verwenden häufig Batch-Hilfsprogramme, um bestimmte Funktionen wie das Sortieren von Daten, das Übertragen von Dateien über FTP, das Laden von Daten in Datenbanken wie DB2, das Entladen von Daten aus Datenbanken usw. auszuführen.

Wenn Sie Ihre Anwendungen zur AWS Mainframe-Modernisierung migrieren, benötigen Sie funktionsgleiche Ersatzdienstprogramme, die dieselben Aufgaben ausführen können wie die, die Sie auf dem Mainframe verwendet haben. Einige dieser Dienstprogramme sind möglicherweise bereits als Teil der Runtime-Engines für die AWS Mainframe-Modernisierung verfügbar, aber wir bieten die folgenden Ersatz-Utilities an:

- M2SFTP — ermöglicht die sichere Dateiübertragung mithilfe des SFTP-Protokolls.
- M2WAIT — wartet eine bestimmte Zeit, bevor mit dem nächsten Schritt in einem Batch-Job fortgefahren wird.
- TXT2PDF - konvertiert Textdateien in das PDF-Format.
- M2DFUTIL — bietet Sicherungs-, Wiederherstellungs-, Lösch- und Kopierfunktionen für Datensätze, die der Unterstützung durch das Mainframe-Hilfsprogramm ADRDSSU ähneln.
- M2RUNCMD — ermöglicht die direkte Ausführung von Micro Focus-Befehlen, -Skripten und Systemaufrufen von JCL aus.

Wir haben diese Batch-Dienstprogramme auf der Grundlage von Kundenfeedback entwickelt und so konzipiert, dass sie dieselben Funktionen wie die Mainframe-Utilities bieten. Ziel ist es, Ihren Übergang von der Mainframe-Modernisierung zur AWS Mainframe-Modernisierung so reibungslos wie möglich zu gestalten.

### Themen

- [Binärer Standort](#)
- [M2SFTP-Batch-Hilfsprogramm](#)
- [M2WAIT Batch-Hilfsprogramm](#)
- [TXT2PDF Batch-Hilfsprogramm](#)

- [M2DFUTIL Batch-Hilfsprogramm](#)
- [M2RUNCMD Batch-Hilfsprogramm](#)

## Binärer Standort

Diese Dienstprogramme sind auf den Produkten Micro Focus Enterprise Developer (ED) und Micro Focus Enterprise Server (ES) vorinstalliert. Sie finden sie für alle Varianten von ED und ES an der folgenden Stelle:

- 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`

## M2SFTP-Batch-Hilfsprogramm

M2SFTP ist ein JCL-Hilfsprogramm, das entwickelt wurde, um sichere Dateiübertragungen zwischen Systemen mithilfe des Secure File Transfer Protocol (SFTP) durchzuführen. Das Programm verwendet den Putty SFTP-Client, um die eigentlichen Dateiübertragungen `psftp` durchzuführen. Das Programm funktioniert ähnlich wie ein Mainframe-FTP-Hilfsprogramm und verwendet Benutzer- und Kennwortauthentifizierung.

### Note

Die Authentifizierung mit öffentlichen Schlüsseln wird nicht unterstützt.

Um Ihre Mainframe-FTP-JCLs zur Verwendung von SFTP zu konvertieren, wechseln Sie zu.  
PGM=FTP PGM=M2SFTP

### Themen

- [Unterstützte Plattformen](#)
- [Installieren von Abhängigkeiten](#)
- [Konfigurieren Sie M2SFTP für Mainframe Modernization Managed AWS](#)
- [M2SFTP für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren \(einschließlich 2.0\) AppStream](#)

- [Beispiel für JCLs](#)
- [Befehlsreferenz für den Putty SFTP \(PSFTP\) -Client](#)
- [Nächste Schritte](#)

## Unterstützte Plattformen

Sie können M2SFTP auf jeder der folgenden Plattformen verwenden:

- AWS Mainframe-Modernisierung, von Micro Focus gemanagt
- Micro Focus Runtime (auf Amazon EC2)
- Alle Varianten der Produkte Micro Focus Enterprise Developer (ED) und Micro Focus Enterprise Server (ES).

## Installieren von Abhängigkeiten

Um den Putty SFTP-Client unter Windows zu installieren

- Laden Sie den [PuTTY SFTP-Client](#) herunter und installieren Sie ihn.

Um den Putty SFTP-Client unter Linux zu installieren:

- Führen Sie den folgenden Befehl aus, um den Putty SFTP-Client zu installieren:

```
sudo yum -y install putty
```

## Konfigurieren Sie M2SFTP für Mainframe Modernization Managed AWS

Wenn Ihre migrierten Anwendungen auf AWS Mainframe Modernization Managed laufen, müssen Sie M2SFTP wie folgt konfigurieren.

- Stellen Sie die entsprechenden Micro Focus Enterprise Server-Umgebungsvariablen für MFFTP ein. Hier sind einige Beispiele:
  - MFFTP\_TEMP\_DIR
  - MFFTP\_SENDEOL

- MFFTP\_TIME
- MFFTP\_ABEND

Sie können so wenige oder so viele dieser Variablen festlegen, wie Sie möchten. Sie können sie mit der ENVAR DD Anweisung in Ihrer JCL setzen. Weitere Informationen zu diesen Variablen finden Sie unter [MFFTP-Steuervariablen](#) in der Micro Focus-Dokumentation.

Informationen zum Testen Ihrer Konfiguration finden Sie unter [Beispiel für JCLs](#)

## M2SFTP für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren (einschließlich 2.0) AppStream

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon EC2 ausgeführt werden, konfigurieren Sie M2SFTP wie folgt.

1. Ändern Sie den [Micro Focus JES-Programmpfad](#) so, dass er den binären Speicherort für Batch-Dienstprogramme enthält. Wenn Sie mehrere Pfade angeben müssen, verwenden Sie Doppelpunkte (:), um Pfade unter Linux und Semikolons (;) unter Windows zu trennen.
  - 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. Stellen Sie die entsprechenden Micro Focus Enterprise Server-Umgebungsvariablen für MFFTP ein. Hier sind einige Beispiele:
  - MFFTP\_TEMP\_DIR
  - MFFTP\_SENDEOL
  - MFFTP\_TIME
  - MFFTP\_ABEND

Sie können so wenige oder so viele dieser Variablen setzen, wie Sie möchten. Sie können sie mit der ENVAR DD Anweisung in Ihrer JCL setzen. Weitere Informationen zu diesen Variablen finden Sie unter [MFFTP-Steuervariablen](#) in der Micro Focus-Dokumentation.

Informationen zum Testen Ihrer Konfiguration finden Sie unter [Beispiel für JCLs](#)

## Beispiel für JCLs

Um die Installation zu testen, können Sie eine der folgenden JCL-Beispieldateien verwenden.

### M2SFTP1.jcl

Diese JCL zeigt, wie man M2SFTP aufruft, um eine Datei an einen Remote-SFTP-Server zu senden. Beachten Sie die Umgebungsvariablen, die in der Anweisung festgelegt sind. ENVVAR DD

```
//M2SFTP1 JOB 'M2SFTP1',CLASS=A,MSGCLASS=X,TIME=1440
/**
/** Copyright Amazon.com, Inc. or its affiliates.*
/** All Rights Reserved.*
/**
/**-----**
/** Sample SFTP JCL step to send a file to SFTP server*
/**-----**
/**
//STEP01 EXEC PGM=M2SFTP,
//          PARM='127.0.0.1 (EXIT=99 TIMEOUT 300)'
/**
//SYSFTPD  DD  *
RECFM FB
LRECL 80
SBSENDEOL CRLF
MBSENDEOL CRLF
TRAILINGBLANKS FALSE
/*
//NETRC    DD  *
machine 127.0.0.1 login sftpuser password sftppass
/*
//SYSPRINT DD  SYSOUT=*
//OUTPUT   DD  SYSOUT=*
//STDOUT   DD  SYSOUT=*
//INPUT    DD  *
type a
locsite notrailingblanks
cd files
put 'AWS.M2.TXT2PDF1.PDF' AWS.M2.TXT2PDF1.pdf
put 'AWS.M2.CARDDEMO.CARDDATA.PS' AWS.M2.CARDDEMO.CARDDATA.PS1.txt
quit
/*
//ENVVAR   DD  *
```

```
MFFTP_VERBOSE_OUTPUT=ON
MFFTP_KEEP=N
/*
/**
//
```

## M2SFTP2.jcl

Diese JCL zeigt, wie M2SFTP aufgerufen wird, um eine Datei von einem Remote-SFTP-Server zu empfangen. Beachten Sie die in der Anweisung festgelegten Umgebungsvariablen. 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**

Wir empfehlen dringend, FTP-Anmeldeinformationen in einer NETRC-Datei zu speichern und den Zugriff nur auf autorisierte Benutzer zu beschränken.

## Befehlsreferenz für den Putty SFTP (PSFTP) -Client

Der PSFTP-Client unterstützt nicht alle FTP-Befehle. Die folgende Liste zeigt alle Befehle, die PSFTP unterstützt.

Befehl	Beschreibung
!	Führen Sie einen lokalen Befehl aus
Tschüss	Beenden Sie Ihre SFTP-Sitzung
cd	Ändern Sie Ihr Remote-Arbeitsverzeichnis
chmod	Dateiberechtigungen und Modi ändern
close	Beenden Sie Ihre SFTP-Sitzung, aber beenden Sie PSFTP nicht
Entf	Dateien auf dem Remoteserver löschen
dir	Entfernte Dateien auflisten
exit	Beenden Sie Ihre SFTP-Sitzung
get	Laden Sie eine Datei vom Server auf Ihren lokalen Computer herunter
help	Gib Hilfe
lcd	Lokales Arbeitsverzeichnis ändern
lpwd	Lokales Arbeitsverzeichnis drucken
ls	Entfernte Dateien auflisten

Befehl	Beschreibung
mget	Laden Sie mehrere Dateien gleichzeitig herunter
mkdir	Erstellen Sie Verzeichnisse auf dem Remoteserver
mput	Laden Sie mehrere Dateien gleichzeitig hoch
mv	Datei (en) auf dem Remoteserver verschieben oder umbenennen
geöffnet	Mit einem Host verbinden
put	Laden Sie eine Datei von Ihrem lokalen Computer auf den Server hoch
pwd	Drucken Sie Ihr Remote-Arbeitsverzeichnis
quit	Beenden Sie Ihre SFTP-Sitzung
bereuen	Setzen Sie das Herunterladen von Dateien fort
ren	Datei (en) auf dem Remoteserver verschieben oder umbenennen
wiedergeben	Fahren Sie mit dem Hochladen von Dateien fort
rm	Löschen Sie Dateien auf dem Remoteserver
rmdir	Entfernen Sie Verzeichnisse auf dem Remoteserver

## Nächste Schritte

Um Dateien mithilfe von SFTP in Amazon Simple Storage Service hoch- und herunterzuladen, können Sie M2SFTP in Verbindung mit dem verwenden AWS Transfer Family, wie in den folgenden Blogbeiträgen beschrieben.

- [Verwenden Sie logische AWS SFTP-Verzeichnisse, um einen einfachen Datenverteilungsservice zu erstellen](#)
- [Aktivieren Sie die Kennwortauthentifizierung für die Verwendung AWS Transfer for SFTP AWS Secrets Manager](#)

## M2WAIT Batch-Hilfsprogramm

M2WAIT ist ein Mainframe-Hilfsprogramm, mit dem Sie eine Wartezeit in Ihren JCL-Skripten einführen können, indem Sie eine Zeitdauer in Sekunden, Minuten oder Stunden angeben. Sie können M2WAIT direkt von JCL aus aufrufen, indem Sie die Zeit, die Sie warten möchten, als Eingabeparameter übergeben. Intern ruft das M2WAIT-Programm das von Micro Focus bereitgestellte Modul auf C\$SLEEP, um auf eine bestimmte Zeit zu warten.

### Note

Sie können Micro Focus-Aliase verwenden, um das zu ersetzen, was Sie in Ihren JCL-Skripten haben. Weitere Informationen finden Sie unter [JES-Alias](#) in der Micro Focus-Dokumentation.

## Themen

- [Unterstützte Plattformen](#)
- [Konfigurieren Sie M2WAIT für AWS Mainframe Modernization Managed](#)
- [M2WAIT für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren \(einschließlich 2.0\) AppStream](#)
- [Beispiel JCL](#)

## Unterstützte Plattformen

Sie können M2WAIT auf jeder der folgenden Plattformen verwenden:

- AWS Mainframe-Modernisierung, von Micro Focus gemanagt
- Micro Focus Runtime (auf Amazon EC2)
- Alle Varianten der Produkte Micro Focus Enterprise Developer (ED) und Micro Focus Enterprise Server (ES).

## Konfigurieren Sie M2WAIT für AWS Mainframe Modernization Managed

Wenn Ihre migrierten Anwendungen auf AWS Mainframe Modernization Managed laufen, müssen Sie M2WAIT wie folgt konfigurieren.

- Verwenden Sie das Programm M2WAIT in Ihrer JCL, indem Sie die Eingabeparameter übergeben, wie unter gezeigt. [Beispiel JCL](#)

## M2WAIT für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren (einschließlich 2.0) AppStream

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon EC2 ausgeführt werden, konfigurieren Sie M2WAIT wie folgt.

1. Ändern Sie den [Micro Focus JES-Programmpfad](#) so, dass er den binären Speicherort für Batch-Dienstprogramme enthält. Wenn Sie mehrere Pfade angeben müssen, verwenden Sie Doppelpunkte (:), um Pfade unter Linux und Semikolons (;) unter Windows zu trennen.
  - 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. Verwenden Sie das Programm M2WAIT in Ihrer JCL, indem Sie den Eingabeparameter übergeben, wie unter gezeigt. [Beispiel JCL](#)

## Beispiel JCL

Um die Installation zu testen, können Sie das M2WAIT1.jcl Programm verwenden.

Diese Beispiel-JCL zeigt, wie man M2WAIT aufruft und ihm mehrere verschiedene Dauern übergibt.

```
//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=*  
//*  
//
```

## TXT2PDF Batch-Hilfsprogramm

TXT2PDF ist ein Mainframe-Hilfsprogramm, das häufig zum Konvertieren einer Textdatei in eine PDF-Datei verwendet wird. Dieses Hilfsprogramm verwendet denselben Quellcode für TXT2PDF (z/OS-Freeware). Wir haben es so geändert, dass es unter der Micro Focus-Laufzeitumgebung von AWS Mainframe Modernization läuft.

### Themen

- [Unterstützte Plattformen](#)
- [Konfigurieren Sie TXT2PDF für AWS Mainframe Modernization Managed](#)
- [TXT2PDF für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren \(einschließlich 2.0\) AppStream](#)
- [Beispiel JCL](#)
- [Modifikationen](#)
- [Referenzen](#)

## Unterstützte Plattformen

Sie können TXT2PDF auf jeder der folgenden Plattformen verwenden:

- AWS Mainframe-Modernisierung, von Micro Focus gemanagt
- Micro Focus Runtime (auf Amazon EC2)
- Alle Varianten der Produkte Micro Focus Enterprise Developer (ED) und Micro Focus Enterprise Server (ES).

## Konfigurieren Sie TXT2PDF für AWS Mainframe Modernization Managed

Wenn Ihre migrierten Anwendungen auf AWS Mainframe Modernization Managed laufen, konfigurieren Sie TXT2PDF wie folgt.

- Erstellen Sie eine REXX EXEC-Bibliothek namens `AWS.M2.REXX.EXEC`. Laden Sie diese [REXX-Module](#) herunter und kopieren Sie sie in die Bibliothek.
  - `TXT2PDF.rex`- TXT2PDF z/OS Freeware (modifiziert)
  - `TXT2PDFD.rex`- TXT2PDF z/OS Freeware (unverändert)
  - `TXT2PDFX.rex`- TXT2PDF z/OS Freeware (modifiziert)
  - `M2GETOS.rex`- Um den Betriebssystemtyp zu überprüfen (Windows oder Linux)

Informationen zum Testen Ihrer Konfiguration finden Sie unter [Beispiel JCL](#).

## TXT2PDF für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren (einschließlich 2.0) AppStream

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon EC2 ausgeführt werden, konfigurieren Sie TXT2PDF wie folgt.

1. Stellen Sie die Micro Focus-Umgebungsvariable `MFREXX_CHARSET` auf den entsprechenden Wert ein, z. B. „“ für ASCII-Daten. A

### Important

Die Eingabe eines falschen Werts kann zu Problemen bei der Datenkonvertierung (von EBCDIC nach ASCII) führen, sodass die resultierende PDF-Datei nicht lesbar oder funktionsunfähig ist. `MFREXX_CHARSETMF_CHARSET`Wir empfehlen eine entsprechende Einstellung.

2. Ändern Sie den [Micro Focus JES-Programmpfad](#) so, dass er den binären Speicherort für Batch-Dienstprogramme enthält. Wenn Sie mehrere Pfade angeben müssen, verwenden Sie Doppelpunkte (:), um Pfade unter Linux und Semikolons (;) unter Windows zu trennen.
  - 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. Erstellen Sie eine REXX EXEC-Bibliothek namens AWS.M2.REXX.EXEC` Laden Sie diese [REXX-Module](#) herunter und kopieren Sie sie in die Bibliothek.
  - TXT2PDF .rex- TXT2PDF z/OS Freeware (modifiziert)
  - TXT2PDFD .rex- TXT2PDF z/OS Freeware (unverändert)
  - TXT2PDFX .rex- TXT2PDF z/OS Freeware (modifiziert)
  - M2GETOS .rex- Um den Betriebssystemtyp zu überprüfen (Windows oder Linux)

Informationen zum Testen Ihrer Konfiguration finden Sie unter [Beispiel JCL](#).

## Beispiel JCL

Um die Installation zu testen, können Sie eine der folgenden JCL-Beispieldateien verwenden.

Txt2PDF1.jcl

Diese JCL-Beispieldatei verwendet einen DD-Namen für die TXT2PDF-Konvertierung.

```
//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

Diese Beispiel-JCL verwendet einen DSN-Namen für die TXT2PDF-Konvertierung.

```
//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=*
/**
//

```

## Modifikationen

Damit das TXT2PDF-Programm auf der Laufzeitumgebung AWS Mainframe Modernization Micro Focus ausgeführt werden kann, haben wir die folgenden Änderungen vorgenommen:

- Änderungen am Quellcode, um die Kompatibilität mit der Micro Focus REXX-Laufzeit sicherzustellen
- Änderungen, um sicherzustellen, dass das Programm sowohl unter Windows- als auch unter Linux-Betriebssystemen ausgeführt werden kann
- Änderungen zur Unterstützung sowohl der EBCDIC- als auch der ASCII-Laufzeit

## Referenzen

TXT2PDF-Referenzen und Quellcode:

- [Konverter von Text zu PDF](#)
- [z/OS Freeware TCP/IP- und E-Mail-Tools](#)
- [TXT2PDF-Benutzerreferenzhandbuch](#)

## M2DFUTIL Batch-Hilfsprogramm

M2DFUTIL ist ein JCL-Hilfsprogramm, das Sicherungs-, Wiederherstellungs-, Lösch- und Kopierfunktionen für Datensätze bereitstellt, ähnlich der Unterstützung durch das Mainframe-Hilfsprogramm ADRDSSU. Dieses Programm behält viele der SYSIN-Parameter von ADRDSSU bei, was die Migration zu diesem neuen Hilfsprogramm vereinfacht.

Themen

- [Unterstützte Plattformen](#)
- [Anforderungen an die Plattform](#)
- [Geplante future Unterstützung](#)
- [Standorte von Vermögenswerten](#)
- [Konfiguration von M2DFUTIL oder AWS Mainframe Modernization Runtime auf Amazon EC2 \(einschließlich 2.0\) AppStream](#)
- [Allgemeine Syntax](#)
- [Beispiel für JCLs](#)

## Unterstützte Plattformen

Sie können M2DFUTIL auf jeder der folgenden Plattformen verwenden:

- Micro Focus ES unter Windows (64 Bit und 32 Bit)
- Micro Focus ES auf Linux (64 Bit)

## Anforderungen an die Plattform

M2DFUTIL hängt davon ab, dass ein Skript aufgerufen wird, um einen Test mit regulären Ausdrücken durchzuführen. Unter Windows müssen Sie Windows Services for Linux (WSL) installieren, damit dieses Skript ausgeführt werden kann.

## Geplante future Unterstützung

Zu den Funktionen, die derzeit nicht über das Mainframe-Hilfsprogramm ADRDSSU verfügbar sind, aber in future verfügbar sein werden, gehören:

- M2 verwaltet
- VSAM
- COPY-Unterstützung für das Umbenennen von Dateinamen
- RENAME-Unterstützung für RESTORE
- Mehrfach INCLUDE und EXCLUDE
- BY-Klausel für die Unterauswahl nach DSORG, CREDIT, EXPDT
- MWAIT-Klausel zur Wiederholung von Fehlern in der Warteschlange
- S3-Speicherunterstützung für DUMP/RESTORE

## Standorte von Vermögenswerten

Das Lademodul für dieses Hilfsprogramm wird M2DFUTIL .so unter Linux und M2DFUTIL .dll Windows aufgerufen. Dieses Lademodul befindet sich an den folgenden Orten:

- 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

Das für das Testen regulärer Ausdrücke verwendete Skript wird aufgerufen `compare.sh`. Dieses Skript befindet sich an den folgenden Orten:

- Linux: /opt/aws/m2/microfocus/utilities/scripts
- Windows (32 Bit): C:\AWS\M2\MicroFocus\Utilities\scripts

## Konfiguration von M2DFUTIL oder AWS Mainframe Modernization Runtime auf Amazon EC2 (einschließlich 2.0) AppStream

Konfigurieren Sie Ihre Enterprise Server-Region wie folgt:

- Fügen Sie die folgenden Variablen in [ES-Environment] hinzu
  - M2DFUTILS\_BASE\_LOC- Der Standardspeicherort für die DUMP-Ausgabe
  - M2DFUTILS\_SCRIPTPATH— Der unter Asset Locations dokumentierte Speicherort des `compare.sh` Skripts
  - M2DFUTILS\_VERBOSE- [AUSFÜHRLICH oder NORMAL]. Dies steuert den Detaillierungsgrad der Ausgabe SYSPRINT
- Stellen Sie sicher, dass der Pfad zum Lademodul der JES\`Configuration\JES Program Path` Einstellung hinzugefügt wurde
- Stellen Sie sicher, dass die Skripts im Verzeichnis der Dienstprogramme über Ausführungsberechtigungen verfügen. In der Linux-Umgebung können Sie mit dem `chmod +x <script name>` Befehl eine Ausführungsberechtigung hinzufügen

### Allgemeine Syntax

#### DUMP

Bietet die Möglichkeit, Dateien vom aktuellen katalogisierten Speicherort an einen Backup-Speicherort zu kopieren. Bei diesem Speicherort muss es sich derzeit um ein Dateisystem handeln.

#### Prozess

DUMP führt Folgendes aus:

1. Erstellen Sie das Zielverzeichnis.
2. Katalogisieren Sie das Zielverzeichnis als PDS-Mitglied.
3. Ermitteln Sie, welche Dateien eingeschlossen werden sollen, indem Sie den INCLUDE-Parameter verarbeiten.
4. Deaktivieren Sie die eingeschlossenen Dateien, indem Sie den EXCLUDE-Parameter verarbeiten.
5. Ermitteln Sie, ob die ausgegebenen Dateien GELÖSCHT werden sollen.
6. Stellen Sie die zu verarbeitenden Dateien in eine Warteschlange.
7. Kopieren Sie die Dateien.

8. Exportieren Sie die katalogisierten DCB-Informationen der kopierten Dateien in eine Nebendatei am Zielort, um future RESTORE-Operationen zu unterstützen.

## Syntax

```
DUMP
TARGET ( TARGET LOCATION ) -
INCLUDE ( DSN. )
[ EXCLUDE ( DSN ) ]
[ CANCEL | IGNORE ]
[ DELETE ]
```

## Erforderliche Parameter

Im Folgenden sind die erforderlichen Parameter für DUMP aufgeführt:

- **SYSPRINT DD NAME**- Um zusätzliche Protokollierungsinformationen zu enthalten
- **TARGET**- Zielort. Dies kann folgendes sein:
  - Vollständiger Pfad des Speicherorts
  - Name des Unterverzeichnisses, das an dem in der Variablen `M2DFUTILS_BASE_LOC` definierten Speicherort erstellt wurde
- **INCLUDE**— Entweder ein einziger benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge
- **EXCLUDE**- Entweder ein einziger benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge

## Optionale Parameter

- **CANCEL** — Stornieren Sie den Vorgang, falls ein Fehler auftritt. Dateien, die verarbeitet wurden, werden beibehalten
- (Standard) **IGNORE** — Ignoriert jeden Fehler und verarbeitet den Vorgang bis zum Ende
- **DELETE** — Wenn kein ENQ-Fehler auftritt, wird die Datei gelöscht und nicht katalogisiert

## DELETE

Ermöglicht das Massentöschung und Entkatalogisieren von Dateien. Dateien werden nicht gesichert.

## Prozess

DELETE führt Folgendes aus:

1. Ermitteln Sie, welche Dateien eingeschlossen werden sollen, indem Sie den INCLUDE-Parameter verarbeiten.
2. Deaktivieren Sie die eingeschlossenen Dateien, indem Sie den EXCLUDE-Parameter verarbeiten.
3. Stellen Sie die zu verarbeitenden Dateien in die Warteschlange. Einstellung der Disposition auf OLD, DELETE, KEEP.

## Syntax

```
DELETE  
INCLUDE ( DSN )  
[ EXCLUDE ( DSN ) ]  
[ CANCEL | IGNORE ]  
[ DELETE ]
```

## Erforderliche Parameter

Im Folgenden sind die erforderlichen Parameter für DELETE aufgeführt:

- **SYSPRINT DD NAME-** Um zusätzliche Protokollierungsinformationen zu enthalten
- **INCLUDE—** Entweder ein einzelner benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge
- **EXCLUDE-** Entweder ein einziger benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge

## Optionale Parameter

- **CANCEL —** Stornieren Sie den Vorgang, falls ein Fehler auftritt. Verarbeitete Dateien werden beibehalten
- **(Standard) IGNORE —** Ignoriert jeden Fehler und verarbeitet den Vorgang bis zum Ende

## RESTORE

Bietet die Möglichkeit, Dateien wiederherzustellen, die zuvor mit DUMP gesichert wurden. Dateien werden am ursprünglichen katalogisierten Speicherort wiederhergestellt, sofern RENAME nicht verwendet wird, um den wiederhergestellten DSNNAME zu ändern.

### Prozess

RESTORE führt Folgendes aus:

1. Überprüfen Sie das Quellverzeichnis.
2. Ermitteln Sie, welche Dateien aufgenommen werden sollen, indem Sie die Katalogexportdatei verarbeiten.
3. Deaktivieren Sie die eingeschlossenen Dateien, indem Sie den EXCLUDE-Parameter verarbeiten.
4. Stellen Sie die zu verarbeitenden Dateien in die Warteschlange.
5. Katalogdateien, die nicht anhand ihrer Exportinformationen katalogisiert wurden.
6. Wenn eine Datei bereits katalogisiert ist und die Exportkataloginformationen identisch sind, ersetzt RESTORE den katalogisierten Datensatz, sofern die Option REPLACE gesetzt ist.

### Syntax

```
RESTORE
SOURCE ( TARGET LOCATION )
INCLUDE ( DSN )
[ EXCLUDE ( DSN ) ]
[ CANCEL | IGNORE ]
[ REPLACE]
```

### Erforderliche Parameter

Im Folgenden sind die erforderlichen Parameter für RESTORE aufgeführt:

- **SYSPRINT DD NAME**— Um zusätzliche Protokollierungsinformationen zu enthalten
- **SOURCE**- Standort der Quelle. Dies kann folgendes sein:
  - Vollständiger Pfad des Speicherorts
  - Name des Unterverzeichnisses, das an dem in der Variablen M2DFUTILS\_BASE\_LOC definierten Speicherort erstellt wurde



- INCLUDE— Entweder ein einziger benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge
- EXCLUDE- Entweder ein einziger benannter DSNAME oder eine gültige Mainframe-DSN-Suchzeichenfolge

### Optionale Parameter

- ABBRECHEN — Stornieren, falls ein Fehler auftritt. Verarbeitete Dateien wurden beibehalten
- (Standard) IGNORE — Ignoriert jeden Fehler und verarbeitet den Vorgang bis zum Ende
- REPLACE — Wenn die wiederherzustellende Datei bereits katalogisiert ist und die Katalogdatensätze identisch sind, ersetzen Sie die katalogisierte Datei

### Beispiel für JCLs

#### DUMP-Job

Dieser Job erstellt ein Unterverzeichnis namens. TESTDUMP Dies ist der Standardspeicherort für Backups, der durch die Variable M2DFUTILS\_BASE\_LOC angegeben wird. Es erstellt eine PDS-Bibliothek für dieses Backup namens. M2DFUTILS . TESTDUMP Die exportierten Katalogdaten werden in einer zeilensequentiellen Datei im Backup-Verzeichnis namens gespeichert. CATDUMP .DAT Alle ausgewählten Dateien werden in dieses Backup-Verzeichnis kopiert.

```
//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
/*
//
```

#### Job LÖSCHEN

Dieser Job löscht alle Dateien aus dem Katalog, die dem INCLUDE-Parameter entsprechen.

```
/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
/*
//
```

## Job WIEDERHERSTELLEN

Dieser Job stellt die Dateien, die dem INCLUDE-Parameter entsprechen, aus dem TESTDUMP Backup-Speicherort wieder her. Katalogisierte Dateien werden ersetzt, wenn die katalogisierte Datei mit der Datei im CATDUMP-Export übereinstimmt und die Option REPLACE angegeben ist.

```
//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
/*
//
```

## M2RUNCMD Batch-Hilfsprogramm

Sie können M2RUNCMD, ein Batch-Hilfsprogramm, verwenden, um Micro Focus-Befehle, -Skripts und Systemaufrufe direkt von JCL aus auszuführen, anstatt sie über ein Terminal oder eine Befehlszeile auszuführen. Die Ausgabe der Befehle wird im Spool-Protokoll des Batch-Jobs protokolliert.

### Themen

- [Unterstützte Plattformen](#)

- [M2RUNCMD für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren \(einschließlich 2.0\) AppStream](#)
- [Beispiel für JCLs](#)

## Unterstützte Plattformen

Sie können M2RUNCMD auf den folgenden Plattformen verwenden:

- Micro Focus Runtime (auf Amazon EC2)
- Alle Varianten der Produkte Micro Focus Enterprise Developer (ED) und Micro Focus Enterprise Server (ES).

## M2RUNCMD für AWS Mainframe Modernization Runtime auf Amazon EC2 konfigurieren (einschließlich 2.0) AppStream

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon EC2 ausgeführt werden, konfigurieren Sie M2RUNCMD wie folgt.

- Ändern Sie den [Micro Focus JES-Programmpfad](#) so, dass er den binären Speicherort für Batch-Dienstprogramme enthält. Wenn Sie mehrere Pfade angeben müssen, verwenden Sie Doppelpunkte (:), um Pfade unter Linux und Semikolons (;) unter Windows zu trennen.
  - 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

## Beispiel für JCLs

Um die Installation zu testen, können Sie eine der folgenden Beispiel-JCLs verwenden.

Führt Scr1.jcl aus

Diese Beispiel-JCL erstellt ein Skript und führt es aus. Im ersten Schritt wird ein Skript mit dem Namen /tmp/TEST\_SCRIPT.sh und Inhalt aus SYSUT1 In-Stream-Daten erstellt. Im zweiten Schritt wird die Ausführungsberechtigung festgelegt und das im ersten Schritt erstellte Skript ausgeführt. Sie können sich auch dafür entscheiden, nur den zweiten Schritt auszuführen, um bereits vorhandene Micro Focus- und Systembefehle auszuführen.

```

//RUNSCL1 JOB 'RUN SCRIPT',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
/*-----*
/* CREATE SCRIPT (LINUX)
/*-----*
//*
//STEP0010 EXEC PGM=IEBGENER
//*
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//*
//SYSUT1 DD *
#!/bin/bash

set -x

## ECHO PATH ENVIRONMENT VARIABLE
echo $PATH

## CLOSE/DISABLE VSAM FILE
casfile -r$ES_SERVER -oc -ed -dACCTFIL

## OPEN/ENABLE VSAM FILE
casfile -r$ES_SERVER -ooi -ee -dACCTFIL

exit $?
/*
//SYSUT2 DD DSN=&&TEMP,
// DISP=(NEW,CATLG,DELETE),
// DCB=(RECFM=LSEQ,LRECL=300,DSORG=PS,BLKSIZE=0)
/*MFE: %PCDSN='/tmp/TEST_SCRIPT.sh'
/*
/*-----*
/* RUN SCRIPT (LINUX)
/*-----*
/*
//STEP0020 EXEC PGM=RUNCMD
//*
//SYSOUT DD SYSOUT=*
//*
//SYSIN DD *
*RUN SCRIPT

```

```
sh /tmp/TEST_SCRIPT.sh
/*
//
```

## SYSOUT

Die Ausgabe des Befehls oder Skripts, das ausgeführt wird, wird in das SYSOUT Protokoll geschrieben. Für jeden ausgeführten Befehl werden der Befehl, die Ausgabe und der Rückgabecode angezeigt.

```
***** 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 *****
```

Führen Sie Cmd1.jcl aus

Diese Beispiel-JCL verwendet RUNCMD, um mehrere Befehle auszuführen.

```
//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
/*
//
```

# AWS Datenreplikation bei der Mainframe-Modernisierung mit Precisely

AWS Mainframe Modernization bietet eine Vielzahl von Amazon Machine Images (AMIs). Diese AMIs ermöglichen die schnelle Bereitstellung von Amazon EC2 EC2-Instances und schaffen so eine maßgeschneiderte Umgebung für die Datenreplikation von Mainframe-Systemen bis hin zur AWS Verwendung von Precisely. Dieses Handbuch enthält die Schritte, die für den Zugriff auf und die Verwendung dieser AMIs erforderlich sind.

## Voraussetzungen

- Stellen Sie sicher, dass Sie Administratorzugriff auf ein AWS Konto haben, mit dem Sie Amazon EC2 EC2-Instances erstellen können.
- Stellen Sie sicher, dass der AWS Mainframe-Modernisierungsservice in der Region verfügbar ist, in der Sie die Amazon EC2 EC2-Instances erstellen möchten. Sehen Sie sich [die Liste der verfügbaren AWS-Services nach Regionen](#) an.
- Identifizieren Sie die Amazon Virtual Private Cloud (Amazon VPC), in der die Amazon EC2 EC2-Instances erstellt werden.
- Stellen Sie beim Erstellen von Amazon EC2 EC2-Instances in einer Amazon VPC sicher, dass die zugehörige Routing-Tabelle über ein Internet-Gateway oder ein NAT-Gateway verfügt.

### Note

Eine erfolgreiche Datenreplikation setzt voraus, dass die AWS EC2-Instance Kommunikationszugriff auf den AWS Marketplace hat. Wenn es ein Verbindungsproblem mit dem AWS Marketplace gibt, schlägt der Replikationsprozess fehl.

## Abonnieren Sie das Amazon Machine Image

Wenn Sie ein AWS Marketplace Marketplace-Produkt abonnieren, können Sie eine Instance über das AMI des Produkts starten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Marketplace Konsole unter <https://console.aws.amazon.com/marketplace>.

2. Wählen Sie Manage subscriptions (Abonnements verwalten).
3. Kopieren Sie den folgenden Link und fügen Sie ihn in die Adressleiste des Browsers ein: <https://aws.amazon.com/marketplace/pp/prodview-en3xrbgzbs3dk>
4. Wählen Sie Continue to Subscribe (Weiter abonnieren) aus.
5. Wenn die Allgemeinen Geschäftsbedingungen akzeptabel sind, wählen Sie Nutzungsbedingungen akzeptieren. Die Bearbeitung des Abonnements kann einige Minuten dauern.
6. Warten Sie, bis die Dankesnachricht angezeigt wird, wie unten gezeigt. Diese Nachricht bestätigt, dass Sie das Produkt erfolgreich abonniert haben.



## AWS Mainframe Modernization service Data Replication with Precisely

Thank you for subscribing to this product! You can now configure your software.

7. Wählen Sie im linken Navigationsbereich die Option Abonnements verwalten aus. In dieser Ansicht werden Ihnen alle Abonnements angezeigt, die Sie abonniert haben.

## Starten Sie die AWS Mainframe Modernization Datenreplikation mit Precisely

1. Öffnen Sie die AWS Marketplace Konsole unter <https://console.aws.amazon.com/marketplace>.
2. Wählen Sie im linken Navigationsbereich die Option Abonnements verwalten aus.
3. Suchen Sie das AMI, das Sie starten möchten, und wählen Sie Neue Instanz starten aus.
4. Wählen Sie unter Region die Region aus, auf der die Liste der zugelassenen Personen aufgeführt ist.
5. Wählen Sie Weiter, um über EC2 zu starten. Mit dieser Aktion gelangen Sie zur Amazon EC2 EC2-Konsole.
6. Geben Sie einen Namen für den Server ein.
7. Wählen Sie einen Instance-Typ aus, der Ihren Leistungs- und Kostenanforderungen für Ihr Projekt entspricht. Der empfohlene Startpunkt für die Instanzgröße istc5.2xLarge.



8. Wählen Sie ein vorhandenes key pair oder erstellen und speichern Sie ein neues. Informationen zu Schlüsselpaaren finden Sie unter [Amazon EC2 EC2-Schlüsselpaare und Linux-Instances](#) im Amazon EC2 EC2-Benutzerhandbuch.
9. Bearbeiten Sie die Netzwerkeinstellungen und wählen Sie die VPC auf der Zulassungsliste und das entsprechende Subnetz aus.
10. Wählen Sie eine bestehende Sicherheitsgruppe oder erstellen Sie eine neue. Für die Datenreplikation mit einer EC2-Instance von Precisely Server wird nicht nur der SSH-Zugriff (standardmäßig auf Port 22) ermöglicht, sondern es ist auch üblich, TCP-Verkehr über den Standardport 2626 zuzulassen.
11. Konfigurieren Sie den Speicher für die Amazon EC2 EC2-Instance.
12. Sehen Sie sich die Zusammenfassung an und wählen Sie Launch Instance aus. Damit der Start erfolgreich ist, muss der Instance-Typ gültig sein. Wenn der Start fehlschlägt, wählen Sie Instance-Konfiguration bearbeiten und wählen Sie einen anderen Instance-Typ.
13. Wenn Sie die Erfolgsmeldung sehen, wählen Sie Connect to instance aus.
14. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
15. Wählen Sie im linken Navigationsbereich unter dem Menü Instances die Option Instances aus.
16. Überprüfen Sie im Hauptbereich den Status Ihrer Instance.

## Eine IAM-Richtlinie erstellen

Um AWS Mainframe Modernization EC2-Instances, die über unsere AWS Marketplace Liste bereitgestellt werden, erfolgreich zu betreiben, müssen Sie eine IAM-Rolle und -Richtlinie konfigurieren. Dieses speziell zugeschnittene IAM-Setup ist nicht optional; es autorisiert Ihre Amazon EC2 EC2-Instances, mit dem Service zu interagieren. AWS Marketplace Die IAM-Rolle und die IAM-Richtlinie ermöglichen es AWS Mainframe Modernization, Nutzungsdaten genau aufzuzeichnen, was für eine genaue Abrechnung unerlässlich ist. Wenn diese Konfiguration nicht implementiert wird, kann dies zu erfolglosen Datenreplikationsversuchen und Betriebsunterbrechungen führen.

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich auf der linken Seite Policies (Richtlinien).
3. Wenn Sie sich zum ersten Mal für Richtlinien entscheiden, wird die Seite Willkommen bei Managed Policies angezeigt. Wählen Sie Get Started.
4. Wählen Sie oben auf der Seite Create policy (Richtlinie erstellen) aus.
5. Wählen Sie im Bereich Policy editor (Richtlinien-Editor) die Option JSON aus.

## 6. Geben Sie die folgende JSON-Richtlinie ein.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": ["aws-marketplace:MeterUsage"],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

## Erstellen einer IAM-Rolle

1. Öffnen Sie die IAM-Konsole unter <https://console.aws.amazon.com/iam/>.
2. Wählen Sie im Navigationsbereich Roles (Rollen) und dann Create role (Rolle erstellen).
3. Wählen Sie im Abschnitt Trusted entity type (Typ der vertrauenswürdigen Entität) die Option AWS service (-Service) aus.
4. Wählen Sie im Abschnitt Anwendungsfall unter Service oder Anwendungsfall Amazon EC2 aus.
5. Wählen Sie Weiter aus.
6. Wählen Sie in der Liste der Richtlinien im Drop-down-Menü Nach Typ filtern die Option Vom Kunden verwaltet aus und geben Sie den Namen der Richtlinie ein, die Sie erstellt haben. Aktivieren Sie das Kontrollkästchen neben dem Namen der Richtlinie.
7. Wählen Sie Weiter aus.
8. Geben Sie einen Namen und optional eine Beschreibung für die Rolle ein.
9. Überprüfen Sie die Vertrauensrichtlinie und die Berechtigungen und wählen Sie dann Rolle erstellen aus.

## Hängen Sie die IAM-Rolle an die Amazon EC2 EC2-Instance an

1. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie Ihre Amazon EC2 EC2-Instance aus.

4. Wählen Sie im Menü Aktionen die Option Sicherheit und dann IAM-Rolle ändern aus.
5. Wählen Sie die Rolle aus, die Sie Ihrer Instance zuordnen möchten, und wählen Sie dann IAM-Rolle aktualisieren aus.

# Integration von Charon

## Einführung in Charon-SSP

1987 veröffentlichte Sun Microsystems den SPARC V7-Prozessor, einen 32-Bit-RISC-Prozessor. 1990 folgte der SPARC V8 — eine Überarbeitung des ursprünglichen SPARC V7, bei der vor allem Hardware-Befehle zum Teilen und Multiplizieren enthalten waren. Die SPARC V8-Prozessoren bildeten die Grundlage für eine Reihe von Servern und Workstations wie die SPARCStation 5, 10 und 20. 1993 folgte auf den SPARC V8 der 64-Bit-Prozessor SPARC V9. Auch dies wurde zur Grundlage für eine Reihe von Servern und Workstations, wie die Enterprise 250 und 450.

Aufgrund der veralteten Hardware und des Mangels an Ersatzteilen oder generalüberholten Teilen ist die Wartung von Software und Systemen, die für diese älteren SPARC-basierten Workstations und Server entwickelt wurden, schwieriger geworden. Um den kontinuierlichen Bedarf an bestimmten end-of-life SPARC-basierten Systemen zu decken, hat Stromasys S.A. die SPARC-Emulatorprodukte der Charon-SSP-Reihe entwickelt. Bei den folgenden Produkten handelt es sich um softwarebasierte, virtuelle Maschinen als Ersatz für die spezifizierten nativen Hardware-SPARC-Systeme. Im Folgenden finden Sie einen allgemeinen Überblick über die emulierten Hardwarefamilien.

Charon-SSP/4M emuliert die folgende SPARC-Hardware:

- Sun-4m-Familie (repräsentiert durch die Sun SparcStation 20): ursprünglich eine Sun-4-Variante mit mehreren Prozessoren, die auf dem MBus-Prozessormodulbus basiert, der in der SPARCServer 600MP-Serie eingeführt wurde. Die Sun-4M-Architektur umfasste später auch Uniprozessorsysteme ohne MBUS wie die SPARCStation 5, die Prozessoren mit SPARC v8-Architektur nutzten. Wird ab SunOS 4.1.2 und von Solaris 2.1 bis Solaris 9 unterstützt. Die Unterstützung für SparcServer 600MP wurde nach Solaris 2.5.1 eingestellt.

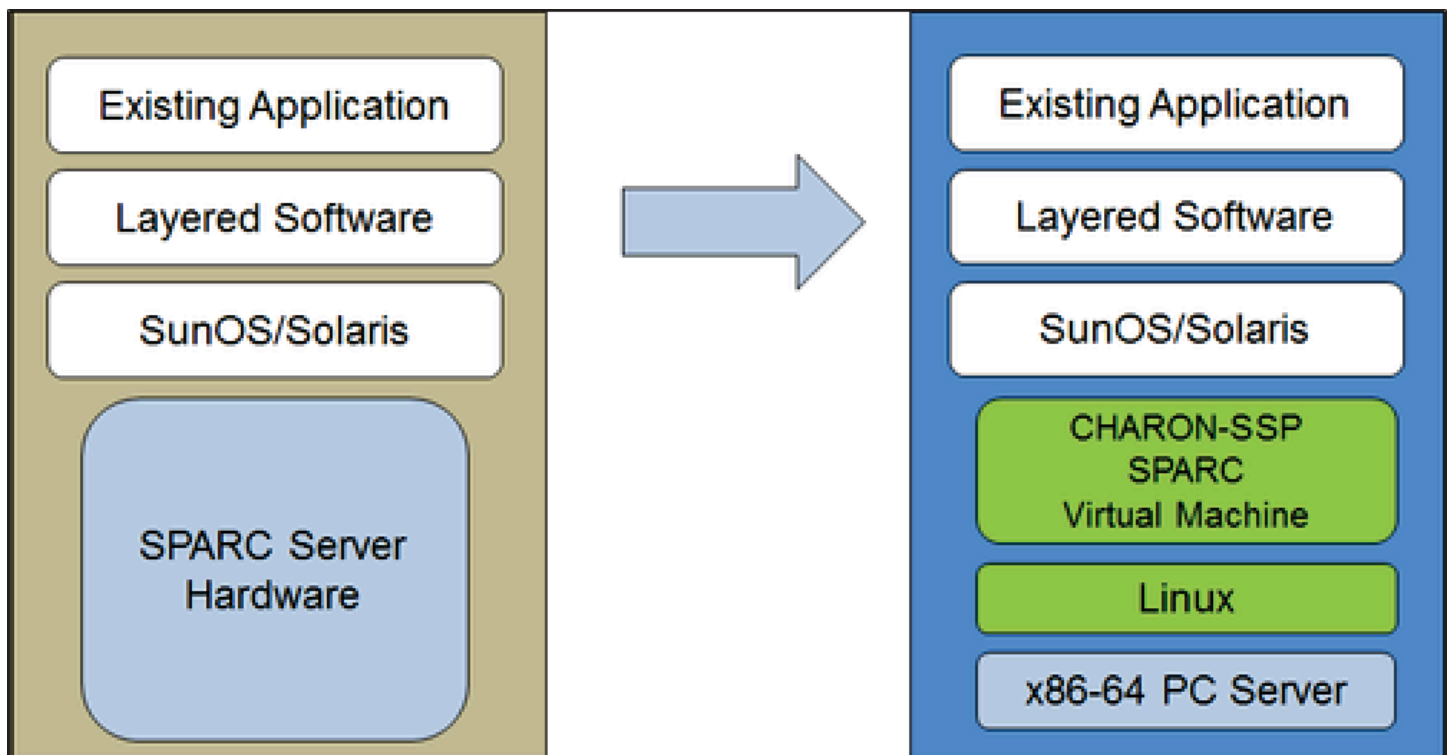
Charon-SSP/4U (+) emuliert die folgende SPARC-Hardware:

- Sun-4U-Familie (vertreten durch Sun Enterprise 450): (U für UltraSPARC) — mit dieser Variante wurden die 64-Bit-SPARC-V9-Prozessorarchitektur und die UPA-Prozessorverbindung eingeführt, die erstmals in der Sun Ultra-Serie verwendet wurden. Wird ab Version 2.5.1 von 32-Bit-Versionen von Solaris unterstützt. Die erste 64-Bit-Solaris-Version für Sun-4U war Solaris 7. Die Unterstützung für UltraSPARC I wurde nach Solaris 9 eingestellt. Solaris 10 unterstützt Sun-4U-Implementierungen von UltraSPARC II bis UltraSPARC IV.

Charon-SSP/4V (+) emuliert die folgende SPARC-Hardware:

- Sun-4V-Familie (repräsentiert durch die SPARC T2 und T4): Diese Variante erweiterte den Sun-4U um Hypervisor-Prozessorvirtualisierung, der im Ultra-SPARC-T1-Multicore-Prozessor eingeführt wurde. Ausgewählte Hardware wurde von Solaris Version 10 ab Version 3/05 HW2 unterstützt (die meisten Modelle — einschließlich der von Charon-SSP emulierten Hardware — erfordern neuere Versionen von Solaris 10). Es werden auch mehrere Solaris 11-Versionen unterstützt.

Die folgende Abbildung zeigt das grundlegende Konzept der Migration von physischer Hardware auf einen Emulator.



Die virtuellen Charon-SSP-Maschinen ermöglichen es Benutzern von Sun- und Oracle SPARC-Computern, ihre native Hardware so zu ersetzen, dass nur geringe oder gar keine Änderungen an der ursprünglichen Systemkonfiguration erforderlich sind. Das bedeutet, dass Sie Ihre Anwendungen und Daten weiterhin ausführen können, ohne auf eine andere Plattform wechseln oder sie portieren zu müssen. Die Charon-SSP-Software läuft auf handelsüblichen Intel 64-Bit-Systemen und gewährleistet so den kontinuierlichen Schutz Ihrer Investition.

Charon-SSP/4U+ unterstützt dieselben virtuellen SPARC-Plattformen wie Charon-SSP/4U und Charon-SSP/4V+ genauso wie Charon-SSP/4V. Die Versionen 4U+ und 4V+ nutzen jedoch die Vorteile der hardwareunterstützten Virtualisierungstechnologie VTx/EPT von Intel und AMD-V/NPT in

modernen CPUs, um eine bessere virtuelle CPU-Leistung zu bieten. Charon-SSP/4U+ und Charon-SSP/4V+ erfordern CPUs mit VT-x/EPT- oder AMD-V/NPT-Unterstützung und müssen auf einem dedizierten Hostsystem installiert werden. Die Ausführung dieser Produktvarianten in einer VM (z. B. auf VMware) wird nicht unterstützt.

### Note

Wenn Sie Charon-SSP/4U+ oder 4V+ in einer Cloud-Umgebung ausführen möchten, wenden Sie sich an Stromasys oder einen Stromasys-VAR, um Ihre Anforderungen zu besprechen.

## Unterstützte Gastbetriebssysteme

Die virtuellen Charon-SSP/4M-Maschinen unterstützen die folgenden Versionen von Gastbetriebssystemen:

- SunOS 4.1.3 — 4.1.4
- Solaris 2.3 auf Solaris 9

Die virtuellen Charon-SSP/4U (+) -Maschinen unterstützen die folgenden Versionen von Gastbetriebssystemen:

- Solaris 2.5.1 bis Solaris 10

Die virtuellen Charon-SSP/4V (+) -Maschinen unterstützen die folgenden Versionen von Gastbetriebssystemen:

- Solaris 10 (ab Update 4, 08/07) und Solaris 11.1 bis Solaris 11.4

Beachten Sie für Charon-SSP/4V (+) Folgendes:

- Für das emulierte SPARC T4 werden folgende Solaris 10-Versionen unterstützt: Oracle Solaris 10 1/13, Oracle Solaris 10 8/11 und Solaris 10 9/10 oder Solaris 10 10/09 mit dem Oracle Solaris 10 8/11 Patch-Set.
- Das emulierte SPARC T4-Modell ist eine Voraussetzung für die Ausführung von Solaris 11.4 im Emulator.
- Solaris-Kernelzonen werden nicht unterstützt.

## Voraussetzungen für die Charon-SSP Cloud-Instanz

Durch Auswahl eines Instanztyps oder einer Instanzform wählen Sie die virtuelle Hardware aus, die für die Charon-SSP-Hostinstanz in der Cloud verwendet werden soll. Daher bestimmt die Auswahl eines Instanztyps oder einer Instanzform die Hardwareeigenschaften der virtuellen Charon-SSP-Host-Hardware (z. B. wie viele CPU-Kerne und wie viel Speicher Ihr virtuelles Charon-Hostsystem haben wird).

### Note

Wenn Sie ein Charon-SSP-Marketplace-Image verwenden, um Ihre Instance zu starten, sind alle Anforderungen an das Linux-Host-Betriebssystem erfüllt.

Die Mindestanforderungen an die Hardware werden unten beschrieben.

Wichtige Punkte zu den Größenrichtlinien:

- Die folgenden Größenrichtlinien, insbesondere in Bezug auf die Anzahl der Host-CPU-Kerne und den Host-Speicher, zeigen die Mindestanforderungen. Jede Bereitstellungssituation muss überprüft werden, und die tatsächliche Hostgröße muss bei Bedarf angepasst werden. Beispielsweise muss die Anzahl der für I/O verfügbaren CPU-Kerne erhöht werden, wenn die Gastanwendungen eine hohe I/O-Last erzeugen. Außerdem ist ein System mit vielen emulierten CPUs in der Regel in der Lage, eine höhere I/O-Last zu erzeugen, sodass die Anzahl der für I/O verfügbaren CPU-Kerne möglicherweise erhöht werden muss. In einer Hyper-Threading-Umgebung muss für optimale Leistung die Anzahl der CPU-Kerne (d. h. reale/physische CPUs) ausreichend sein, um die CPU-Anforderungen der aktiven Emulatoren zu erfüllen. Dadurch wird vermieden, dass sich Threads mit hoher Arbeitslast einen physischen CPU-Kern teilen.
- Die Zuordnung von CPU-Kernen für emulierte CPUs und CPU-Kerne für die I/O-Verarbeitung wird durch die Konfiguration bestimmt. Weitere Informationen dazu und zur Standardzuweisung von CPU-Kernen für die I/O-Verarbeitung finden Sie unter CPU-Konfiguration im allgemeinen Charon-SSP-Benutzerhandbuch.

### Wichtige allgemeine Informationen

- Um eine schnelle Übertragung von Emulatordaten von einer Cloud-Instanz zur anderen zu ermöglichen, wird dringend empfohlen, alle relevanten Emulatordaten auf einem separaten

Festplattenvolume zu speichern, das einfach von der alten Instanz getrennt und an eine neue Instanz angehängt werden kann.

- Stellen Sie sicher, dass Ihre Instanz von Anfang an korrekt dimensioniert ist (überprüfen Sie die Mindestanforderungen unten). Die Charon-SSP-Lizenz für Charon-SSP AL wird erstellt, wenn die Instanz zum ersten Mal gestartet wird. Wenn Sie später zu einer anderen Instanzgröße/einem anderen Instanztyp wechseln und dadurch die Anzahl der CPU-Kerne ändern, wird die Lizenz ungültig und somit verhindert, dass Charon-Instanzen gestartet werden (neue Instanz erforderlich). Wenn Sie planen, die Charon-SSP AL-Instanz im AutoVE-Modus zu verwenden, stellen Sie sicher, dass Sie die AutoVE-Serverinformationen vor dem ersten Start angeben, da andernfalls die öffentlichen Lizenzserver verwendet werden. Die Lizenz für Charon-SSP VE wird auf der Grundlage des Fingerabdrucks erstellt, der auf dem Lizenzserver aufgenommen wurde. Wenn der Lizenzserver direkt auf dem Emulatorhost ausgeführt wird und der Emulatorhost später beispielsweise eine Änderung der Anzahl der CPU-Kerne erfordert, wird die Lizenz ungültig (neue Lizenz und möglicherweise neue Instanz erforderlich).

## Voraussetzungen für die Instanz

Allgemeine CPU-Anforderungen: Charon-SSP unterstützt moderne Prozessoren mit der x86-64-Architektur, die auf Amazon EC2 EC2-Instances basieren.

Mindestanforderungen für Charon-SSP:

- Mindestanzahl von CPU-Kernen des Hostsystems:
  - Mindestens ein CPU-Kern für das Hostbetriebssystem sowie:
  - Für jedes emulierte SPARC-System:
    - Ein CPU-Kern für jede emulierte CPU der Instanz plus:
    - Mindestens ein zusätzlicher CPU-Kern für die I/O-Verarbeitung (mindestens zwei, wenn die Server-JIT-Optimierung verwendet wird). Die Konfigurationsoptionen finden Sie im oben genannten Abschnitt zur CPU-Konfiguration. Standardmäßig weist Charon 1/3 (min. 1; abgerundet) der Anzahl der CPUs, die für den Charon-Host sichtbar sind, der I/O-Verarbeitung zu.
- Minimale Speicheranforderungen:
  - 4 GB RAM oder mehr für das Linux-Host-Betriebssystem. Die tatsächlichen Anforderungen können höher sein und hängen von den Anforderungen der Nicht-Emulator-Dienste ab, die auf



dem Linux-Host ausgeführt werden. Die vorherige Empfehlung von mindestens 2 GB RAM für den Linux-Host wird weiterhin für viele Systeme gelten, aber die steigenden Anforderungen an das Linux-Betriebssystem und die Linux-Anwendungen haben zu der aktualisierten Empfehlung für neue Installationen geführt. Außerdem:

- Für jedes emulierte SPARC-System:
  - Der konfigurierte Speicher der emulierten Instanz plus:
  - 2 GB RAM (6 GB RAM, wenn Server-JIT verwendet wird), um die DIT-Optimierung, Emulatoranforderungen, Laufzeitpuffer, SMP und Grafikumulation zu ermöglichen.
- Wenn Hyperthreading auf modernen x86-64-CPU's aktiviert ist, können zwei Threads auf einem physischen CPU-Kern ausgeführt werden, wodurch dem Host-Betriebssystem zwei logische CPU's zur Verfügung stehen. Deaktivieren Sie nach Möglichkeit Hyperthreading auf dem Charon-SSP-Host. In VMware- und Cloud-Umgebungen ist dies jedoch häufig nicht möglich, oder es ist unklar, ob Hyperthreading verwendet wird oder nicht. Die Hyper-Threading-Option Charon-SSP ermöglicht es Charon-SSP, sich an solche Umgebungen anzupassen. Ausführliche Informationen zur Konfiguration finden Sie im Abschnitt CPU-Konfiguration in Ihrem oben genannten allgemeinen Charon-SSP-Benutzerhandbuch. Bitte beachten Sie: Um eine optimale Leistung zu erzielen, sollten sich Charon-SSP-Threads keinen physischen CPU-Kern teilen. Auf dem Hostsystem sollten genügend physische Kerne verfügbar sein, um die Anforderungen des/der konfigurierten Emulatoren zu erfüllen.
- Eine oder mehrere Netzwerkschnittstellen, je nach Kundenanforderungen.
- Charon-SSP/4U+ und Charon-SSP/4V+ müssen auf physischer Hardware laufen, die Intel VT-x/EPT oder AMD-V/NPT (Baremetal-Instanzen) unterstützt, und können daher nicht in allen Cloud-Umgebungen ausgeführt werden. Bitte überprüfen Sie die Dokumentation Ihres Cloud-Anbieters auf die Verfügbarkeit dieser Hardware. Beachten Sie außerdem die folgenden Punkte:
  - Charon-SSP/4U+ und Charon-SSP/4V+ sind nur verfügbar, wenn ein von Stromasys unterstützter Linux-Kernel verwendet wird.
  - Wenn Sie diese Art von emulierter SPARC-Hardware benötigen, wenden Sie sich an Stromasys oder Ihren Stromasys-VAR, um Ihre Anforderungen im Detail zu besprechen.

# Erstellen und Konfigurieren einer AWS Cloud-Instanz für Charon (neue GUI)

Dieser Abschnitt spiegelt das AWS Management Console im Frühjahr 2022 wider. Wenn Sie immer noch die ältere Konsole verwenden, lesen Sie im Anhang des Charon-SSP AWS Getting Started Handbuchs nach.

## Allgemeine Voraussetzungen

Diese Beschreibung zeigt die grundlegende Einrichtung einer Linux-Instanz in AWS. Es werden keine spezifischen Voraussetzungen aufgeführt. Beachten Sie jedoch je nach Anwendungsfall die folgenden Voraussetzungen:

- Amazon-Konto und AWS Marketplace Abonnements
  - Um eine Linux-Instanz einzurichten AWS, benötigen Sie ein AWS Konto mit Administratorzugriff.
  - Identifizieren Sie die AWS Region, in der Sie Ihre Instanz starten möchten. Stellen Sie sicher, dass die AWS Dienste, die Sie nutzen möchten, in dieser Region verfügbar sind. Siehe [AWS Dienste nach Regionen](#).
  - Identifizieren Sie die VPC und das Subnetz, in denen Sie Ihre Instanz starten möchten.
  - Wenn Ihre Instanz Internetzugang benötigt, stellen Sie sicher, dass die mit Ihrer VPC verknüpfte Routing-Tabelle über ein Internet-Gateway verfügt. Wenn Ihre Instanz VPN-Zugriff auf Ihr lokales Netzwerk benötigt, stellen Sie sicher, dass ein VPN-Gateway verfügbar ist. Die genaue Konfiguration Ihrer VPC und ihrer Subnetze hängt von Ihrem Netzwerkdesign und Ihren Anwendungsanforderungen ab.
  - Um einen bestimmten AWS Marketplace Service zu abonnieren, wählen Sie im die Option AWS Marketplace Marketplace-Abonnements AWS Management Console und dann Abonnements verwalten aus.
  - Suchen Sie nach dem Service, den Sie nutzen möchten, und abonnieren Sie ihn. Nach einem erfolgreichen Abonnement finden Sie das Abonnement im Bereich Abonnements verwalten. Von dort aus können Sie direkt eine neue Instanz starten.
- Die Hardware- und Softwarevoraussetzungen für die Instanz sind je nach der geplanten Nutzung der Instanz unterschiedlich:
  - Option 1: Die Instanz soll als Charon-Emulator-Hostsystem verwendet werden:

- Die genauen Hardware- und Softwarevoraussetzungen, die von der Linux-Instanz erfüllt werden müssen, finden Sie in den Abschnitten zu den Hardware- und Softwarevoraussetzungen im Benutzerhandbuch und/oder im Handbuch Erste Schritte Ihres Charon-Produkts. Das Image, das Sie zum Starten Ihrer Instance verwenden, und der Instance-Typ, den Sie ausgewählt haben, bestimmen die Software und Hardware Ihrer Cloud-Instanz.
- Für den Betrieb emulierter Altsysteme ist eine Charon-Produktlizenz erforderlich. Weitere Informationen entnehmen Sie den Lizenzinformationen in der Dokumentation Ihres Charon-Produkts oder wenden Sie sich an Ihren Stromasys-Vertreter oder an Stromasys VAR.
- Option 2: Die Instanz soll als dedizierter VE-Lizenzserver verwendet werden:
  - Ausführliche Informationen zu den Voraussetzungen finden Sie im VE-Lizenzserver-Handbuch.
- Bestimmte ältere Betriebssysteme, die auf den von Charon-Emulatorprodukten bereitgestellten emulierten Systemen ausgeführt werden können, erfordern eine Lizenz des ursprünglichen Anbieters des Betriebssystems. Der Benutzer ist für alle Lizenzverpflichtungen im Zusammenhang mit dem älteren Betriebssystem verantwortlich und muss die entsprechenden Lizenzen bereitstellen.

## Verwenden von AWS Management Console , um eine neue Instanz zu starten

Um eine neue Instanz zu erstellen

1. Melden Sie sich bei der Amazon EC2 EC2-Konsole an AWS Management Console und öffnen Sie sie unter <https://console.aws.amazon.com/ec2/>.
2. Wählen Sie Launch Instance (Instance starten) aus.
3. Geben Sie einen Namen für die Instance ein.
4. Wählen Sie ein AMI aus. Ein AMI ist ein vorgefertigtes Image, das zum Starten von Cloud-Instanzen verwendet wird. Es umfasst das Betriebssystem und die entsprechende Anwendungssoftware. Die Wahl des AMI hängt davon ab, wie Sie die Instance verwenden möchten:
  - Wenn die Instance als Charon-Emulator-Hostsystem verwendet werden soll, sind mehrere AMI-Optionen möglich:


- Installation des Charon-Hostsystems von einem vorkonfigurierten Charon-Marketplace-Image aus: Sie enthalten das zugrunde liegende Betriebssystem und die vorinstallierte Charon-Software.
- Erkundigen Sie sich bei Ihrem Stromasys-Vertreter, welche Optionen derzeit auf dem Marktplatz Ihres Cloud-Anbieters verfügbar sind.
- Abhängig vom Cloud-Anbieter und den Release-Plänen für Stromasys-Produkte kann es zwei Varianten geben:
  - Automatische Lizenzierung (AL) für die Verwendung mit einem öffentlichen, von Stromasys betriebenen Lizenzserver oder mit einem privaten, vom Kunden betriebenen AutoVE-Lizenzserver
  - Virtuelle Umgebung (VE) zur Verwendung mit einem privaten, vom Kunden betriebenen VE-Lizenzserver
- Installation des Charon-Hostsystems mithilfe einer herkömmlichen Charon-Emulatorinstallation mit den RPM-Paketen für die Charon-Emulator-Installation für Linux:
  - Wählen Sie ein Linux-AMI einer Distribution, die von Ihrem ausgewählten Charon-Produkt und der ausgewählten Charon-Version unterstützt wird. Das Benutzerhandbuch für Ihr Produkt finden Sie auf der Stromasys-Dokumentationsseite.
- Wenn die Instanz als dedizierter VE-Lizenzserver verwendet werden soll, finden Sie die Anforderungen der Linux-Instanz im VE-Lizenzserver-Handbuch in der Lizenzierungsdokumentation.

Nachdem Sie entschieden haben, welches AMI erforderlich ist, wählen Sie ein passendes Linux- oder Charon-Produkt-AMI aus. Wenn Sie das benötigte AMI nicht sehen, wählen Sie [Browse more AMIs](#) aus. Wählen Sie das Linux-AMI aus, das Ihrer geplanten Verwendung der Instance entspricht. Es kann sich dabei um eines der folgenden handeln:

- Ein vorkonfiguriertes Charon VE Marketplace-Image. Der Name des AMI wird die Zeichenfolge „ve“ enthalten.
  - Ein vorkonfiguriertes Charon AL-Marketplace-Image für Automatic Licensing oder AutoVE.
  - Eine Linux-Version, die für die Installation eines RPM-Produkts unterstützt wird.
  - Eine Linux-Version, die für den VE-Lizenzserver unterstützt wird.
5. Auswahl von Instance-Typen Amazon EC2 bietet Instance-Typen mit unterschiedlichen Kombinationen von CPU-, Arbeitsspeicher-, Speicher- und Netzwerkkapazität. Wählen Sie einen

Instance-Typ aus, der den Anforderungen des Charon-Produkts entspricht, das Sie verwenden möchten. Bei einigen Marketplace-Bildern ist die Auswahl an Instance-Typen eingeschränkt.

- Wählen Sie ein vorhandenes key pair aus oder erstellen und speichern Sie ein neues. Wenn Sie ein vorhandenes key pair auswählen, stellen Sie sicher, dass Sie den passenden privaten Schlüssel haben. Andernfalls können Sie keine Verbindung zu Ihrer Instance herstellen.

 Note

Wenn Ihr Managementsystem dies unterstützt, verwenden Sie für RHEL 9.x, Rocky Linux 9.x und Oracle Linux 9.x den SSH-Schlüsseltyp ECDSA oder ED25519. Mit diesen Typen können Sie mithilfe eines SSH-Tunnels eine Verbindung zu diesen Charon-Host-Linux-Systemen herstellen, ohne die standardmäßigen Krypto-Richtlinieneinstellungen auf dem Charon-Host in weniger sichere Einstellungen ändern zu müssen. Dies ist beispielsweise für den Charon-SSP Manager wichtig. Weitere Informationen finden Sie [unter Verwendung systemweiter kryptografischer Richtlinien](#) in der Red Hat-Dokumentation.

- Wählen Sie im Bereich Netzwerkeinstellungen die Option Bearbeiten. Wählen Sie die Einstellungen aus, die Ihrer Umgebung entsprechen.
  - Geben Sie eine VPC an.
  - Geben Sie ein vorhandenes Subnetz an oder erstellen Sie ein neues.
  - Aktivieren oder deaktivieren Sie die automatische Zuweisung einer öffentlichen IP-Adresse zur primären Schnittstelle. Die automatische Zuweisung ist nur möglich, wenn die Instanz über eine einzige Netzwerkschnittstelle verfügt.
  - Weisen Sie eine bestehende oder eine neue benutzerdefinierte Sicherheitsgruppe zu. Die Sicherheitsgruppe muss mindestens SSH den Zugriff auf die Instanz ermöglichen. Alle Ports, die von Anwendungen benötigt werden, die Sie auf der Instance ausführen möchten, müssen ebenfalls zugelassen werden. Sie können die Sicherheitsgruppe jederzeit ändern, nachdem Sie die Instanz erstellt haben.
- Wählen Sie im Bereich Speicher für das Root-Volume (die Systemfestplatte) eine Größe aus, die für Ihre Umgebung geeignet ist. Die empfohlene Mindestgröße der Systemfestplatte für das Linux-System beträgt 30 GiB. Um Speicherplatz für virtuelle Festplattencontainer und andere Speicheranforderungen bereitzustellen, können Sie jetzt oder nach dem Start der Instance weiteren Speicherplatz hinzufügen. Die Größe der Systemfestplatte muss jedoch den Linux-

Systemanforderungen entsprechen, einschließlich aller Anwendungen und Dienstprogramme, die Sie installieren möchten.

**Note**

Wir empfehlen, separate Speichervolumen für Charon-Anwendungsdaten (z. B. Festplatten-Images) zu erstellen. Bei Bedarf können Sie solche Volumens später auf eine andere Instanz migrieren.

- Erweitern Sie den Abschnitt Erweiterte Details, scrollen Sie nach unten und wählen Sie CPU-Optionen angeben aus. Drei davon, die für eine Charon-Emulatorumgebung mit größerer Wahrscheinlichkeit nützlich sind, sind in der folgenden Abbildung als Beispiele dargestellt.



Specify CPU options

Core count

2

Threads per core

2

Number of vCPUs

4

- Bei einem VE-Lizenzserversystem mit einer Version vor 1.1.23 müssen Sie der Instanz die erforderliche IAM-Rolle zuweisen. Es muss sich um eine Rolle handeln, die die `ListUsers` Aktion ermöglicht. Um eine Rolle zuzuweisen, wählen Sie im erweiterten Abschnitt Erweiterte Details entweder eine Rolle unter IAM-Instanzprofil aus oder klicken Sie auf Neues IAM-Profil erstellen. Weitere Informationen finden Sie unter [IAM-Rollen für Amazon EC2](#).
- Wenn Ihre Instance auf einem Charon AWS Marketplace AL-Image basiert und Sie beabsichtigen, die von StromaSYS betriebenen öffentlichen Lizenzserver zu verwenden, müssen Sie die entsprechenden Informationen zur Instanzkonfiguration hinzufügen, bevor Sie die Instance starten.

Geben Sie die Informationen für den AutoVE-Lizenzserver ein, wie in der folgenden Abbildung gezeigt.

Metadata accessible [Info](#)

Enabled

Metadata version [Info](#)

V1 and V2 (token optional)

Metadata response hop limit [Info](#)

Select

Allow tags in metadata [Info](#)

Select

User data [Info](#)

primary\_server=172.31.34.235:8083

User data has already been base64 encoded

Die folgenden Konfigurationsoptionen für Benutzerdaten sind gültig:

- **primary\_server**=<ip-address>[:<port>]
- **backup\_server**=<ip-address>[:<port>]

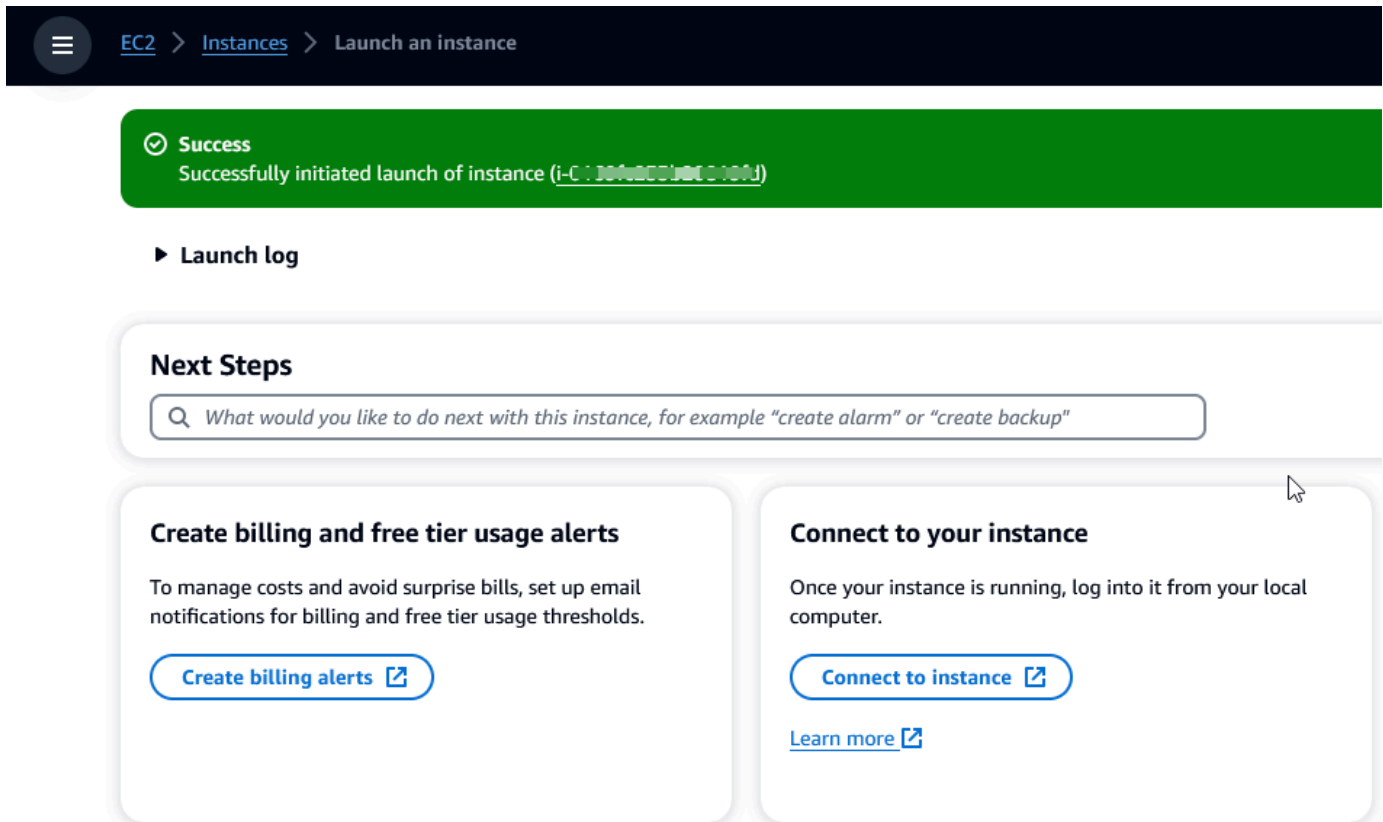
Wo

- <ip-address>steht für die IP-Adresse des Primär- und des Backup-Servers, sofern zutreffend.
- <port>steht für einen nicht standardmäßigen TCP-Port, der für die Kommunikation mit dem Lizenzserver verwendet wird (Standard: TCP/8083).

**Note**

Beim ersten Start muss mindestens ein Lizenzserver konfiguriert werden, um den AutoVE-Modus zu aktivieren. Andernfalls bindet sich die Instanz an einen der öffentlichen Lizenzserver, die von Stromasys betrieben werden.

12. Wählen Sie im Abschnitt Zusammenfassung die Option Launch instance aus. Nach einer Weile wird die folgende Erfolgsmeldung angezeigt:



The screenshot shows the AWS Management Console interface. At the top, a dark navigation bar contains a hamburger menu icon, the text "EC2 > Instances > Launch an instance", and a search icon. Below this is a green success notification bar with a checkmark icon, the text "Success", and "Successfully initiated launch of instance (i-01304600100000000)". Underneath is a "Launch log" section with a right-pointing triangle icon. The "Next Steps" section features a search bar with the placeholder text "What would you like to do next with this instance, for example 'create alarm' or 'create backup'". Below the search bar are two white cards. The left card is titled "Create billing and free tier usage alerts" and contains the text "To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds." with a blue button labeled "Create billing alerts" and an external link icon. The right card is titled "Connect to your instance" and contains the text "Once your instance is running, log into it from your local computer." with a blue button labeled "Connect to instance" and an external link icon, and a link labeled "Learn more" with an external link icon.

13. Wählen Sie in der unteren rechten Ecke des Bildschirms die Option Alle Instanzen anzeigen aus.
14. Um die Details Ihrer Instanz zu sehen, aktivieren Sie das Kontrollkästchen links neben der Zeile, die die Instanz in der Instanztabelle darstellt. Ihre Instanzdetails werden in der unteren Hälfte des Bildschirms angezeigt. Informationen darüber, wie Sie eine Verbindung zu Ihrer Instance herstellen, finden [Sie unter Connect](#) im Amazon EC2 EC2-Benutzerhandbuch.



# AWS Neuplattform-Modernisierung von Mainframes mit NTT DATA

AWS Mainframe Modernization bietet eine Vielzahl von Amazon Machine Images (AMIs). Diese AMIs ermöglichen die schnelle Bereitstellung von Amazon EC2 EC2-Instances und schaffen eine maßgeschneiderte Umgebung für das Rehosting und die Neuplattformierung von Mainframe-Anwendungen mithilfe von NTT Data. AWS Dieses Handbuch enthält die Schritte, die für den Zugriff auf und die Verwendung dieser AMIs erforderlich sind.

## Voraussetzungen

- Stellen Sie sicher, dass Sie Administratorzugriff auf ein AWS Konto haben, mit dem Sie Amazon EC2 EC2-Instances erstellen können.
- Stellen Sie sicher, dass der AWS Mainframe-Modernisierungsservice in der Region verfügbar ist, in der Sie die Amazon EC2 EC2-Instances erstellen möchten. Sehen Sie sich [die Liste der verfügbaren AWS-Services nach Regionen](#) an.
- Identifizieren Sie die Amazon VPC, in der Sie die Amazon EC2 EC2-Instances erstellen möchten.

## Abonnieren Sie das Amazon Machine Image

Wenn Sie ein AWS Marketplace Marketplace-Produkt abonnieren, können Sie eine Instance über das AMI des Produkts starten.

1. Melden Sie sich bei der an AWS Management Console und öffnen Sie die AWS Marketplace Konsole unter <https://console.aws.amazon.com/marketplace>.
2. Wählen Sie Manage subscriptions (Abonnements verwalten).
3. Kopieren Sie den folgenden Link und fügen Sie ihn in die Adressleiste des Browsers ein: <https://aws.amazon.com/marketplace/pp/prodview-eg227ymlsdx2>
4. Wählen Sie Continue to Subscribe (Weiter abonnieren) aus.
5. Wenn die Allgemeinen Geschäftsbedingungen akzeptabel sind, wählen Sie Bedingungen akzeptieren. Die Bearbeitung des Abonnements kann einige Minuten dauern.
6. Warten Sie, bis eine Dankesnachricht erscheint. Diese Nachricht bestätigt, dass Sie das Produkt erfolgreich abonniert haben.

7. Wählen Sie im linken Navigationsbereich die Option Abonnements verwalten aus. In dieser Ansicht werden Ihnen alle Ihre Abonnements angezeigt.

## Starten Sie die AWS Mainframe Modernization Replatform mit der NTT DATA-Instanz

1. [Öffnen Sie die AWS Marketplace Konsole unter https://console.aws.amazon.com/marketplace](https://console.aws.amazon.com/marketplace).
2. Wählen Sie im linken Navigationsbereich die Option Abonnements verwalten aus.
3. Suchen Sie das AMI, das Sie starten möchten, und wählen Sie Neue Instanz starten aus.
4. Wählen Sie unter Region die Region aus, auf der die Liste der zugelassenen Personen aufgeführt ist.
5. Wählen Sie Weiter, um über EC2 zu starten. Mit dieser Aktion gelangen Sie zur Amazon EC2 EC2-Konsole.
6. Geben Sie einen Namen für den Server ein.
7. Wählen Sie einen Instance-Typ, der Ihren Leistungs- und Kostenanforderungen für Ihr Projekt entspricht. Der empfohlene Startpunkt für die Instanzgröße ist `c5.2xLarge`.
8. Wählen Sie ein vorhandenes key pair oder erstellen und speichern Sie ein neues. Informationen zu Schlüsselpaaren finden Sie unter [Amazon EC2 EC2-Schlüsselpaare und Linux-Instances](#) im Amazon EC2 EC2-Benutzerhandbuch.
9. Bearbeiten Sie die Netzwerkeinstellungen und wählen Sie die VPC auf der Zulassungsliste und das entsprechende Subnetz aus.
10. Wählen Sie eine bestehende Sicherheitsgruppe oder erstellen Sie eine neue. Wenn es sich um eine Enterprise Server Amazon EC2 EC2-Instance handelt, ist es in der Regel üblich, TCP-Verkehr zu den Ports 86 und 10086 zuzulassen, um die Micro Focus-Konfiguration zu verwalten.
11. Konfigurieren Sie den Speicher für die Amazon EC2 EC2-Instance.
12. Sehen Sie sich die Zusammenfassung an und wählen Sie Launch Instance aus. Damit der Start erfolgreich ist, muss der Instance-Typ gültig sein. Wenn der Start fehlschlägt, wählen Sie Instance-Konfiguration bearbeiten und wählen Sie einen anderen Instance-Typ.
13. Wenn Sie die Erfolgsmeldung sehen, wählen Sie Connect to instance aus.
14. Öffnen Sie die Amazon EC2-Konsole unter <https://console.aws.amazon.com/ec2/>.
15. Wählen Sie im linken Navigationsbereich unter dem Menü Instances die Option Instances aus.
16. Überprüfen Sie im Hauptbereich den Status Ihrer Instance.



- NTT DATA ist sicher UniKix
- Zentraler Manager von NTT DATA UniKix

Der `software` Ordner, der im vorherigen Bild angezeigt wird, enthält die Binärdateien für die oben aufgeführten Komponenten.

Nachdem Sie die Amazon EC2 EC2-Instance erfolgreich validiert haben, beginnen Sie mit der Nutzung von AWS Mainframe Modernization Replatform mit NTT DATA, indem Sie der NTT Data-Dokumentation folgen.

# Anwendungen bei der AWS Mainframe-Modernisierung

Wenn Sie mit der AWS Mainframe-Modernisierung noch nicht vertraut sind, finden Sie zunächst die folgenden Themen:

- [Was ist AWS Mainframe-Modernisierung?](#)
- [Einrichtung der AWS Mainframe-Modernisierung](#)
- [Tutorial: Verwaltete Laufzeit für AWS Blu Age](#)
- [Tutorial: Verwaltete Laufzeit für Micro Focus](#)

Eine Anwendung im Rahmen von AWS Mainframe Modernization enthält einen migrierten Mainframe-Workload. Die Anwendung entspricht einem Workload auf dem Mainframe und ist einer Laufzeitumgebung zugeordnet. Sie können Batchdateien und Datensätze zu Anwendungen hinzufügen und Anwendungen überwachen, während sie ausgeführt werden. Sie erstellen AWS Mainframe-Modernisierungsanwendungen für jeden Workload, den Sie migrieren. Wenn Sie eine AWS Mainframe-Modernisierungsanwendung erstellen, geben Sie bei der Erstellung die Engine an, auf der die Anwendung ausgeführt wird. Wählen Sie AWS Blu Age, wenn Sie das automatisierte Refactoring-Muster verwenden, und wählen Sie Micro Focus, wenn Sie das Replatforming-Muster verwenden.

## Themen

- [Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS](#)
- [Stellen Sie eine AWS Mainframe-Modernisierungsanwendung bereit](#)
- [Aktualisieren Sie eine AWS Mainframe-Modernisierungsanwendung](#)
- [Löschen Sie eine AWS Mainframe-Modernisierungsanwendung aus einer Umgebung](#)
- [Löschen Sie eine AWS Mainframe-Modernisierungsanwendung](#)
- [Reichen Sie Batch-Jobs für AWS Mainframe-Modernisierungsanwendungen ein](#)
- [Importieren Sie Datensätze für AWS Mainframe-Modernisierungsanwendungen](#)
- [Verwalten Sie Transaktionen für AWS Mainframe-Modernisierungsanwendungen](#)
- [AWS Ressourcen für eine migrierte Anwendung erstellen](#)
- [Die verwaltete Anwendung konfigurieren](#)
- [AWS Anwendungsdefinitionsreferenz für Mainframe-Modernisierung](#)
- [AWS Referenz zur Definition von Datensätzen zur Mainframe-Modernisierung](#)

# Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, um eine AWS Mainframe-Modernisierungsanwendung zu erstellen.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Erstellen einer Anwendung

So erstellen Sie eine Anwendung

1. [Öffnen Sie die AWS Mainframe-Modernisierungskonsole unter https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Anwendung erstellen möchten.
3. Wählen Sie auf der Seite Applications die Option Create application aus.
4. Geben Sie auf der Seite Basisinformationen an, im Abschnitt Name und Beschreibung einen Namen für die Anwendung ein.
5. (Optional) Geben Sie im Feld Anwendungsbeschreibung eine Beschreibung für die Anwendung ein. Diese Beschreibung kann Ihnen und anderen Benutzern helfen, den Zweck der Anwendung zu identifizieren.
6. Wählen Sie im Abschnitt Engine-Typ Blu Age für automatisiertes Refactoring oder Micro Focus für Replatforming.
7. Wählen Sie im Abschnitt KMS-Schlüssel die Option Verschlüsselungseinstellungen anpassen aus, wenn Sie einen vom Kunden verwalteten Schlüssel verwenden möchten. AWS KMS Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für den AWS Mainframe-Modernisierungsservice](#).

### Note

Standardmäßig verschlüsselt AWS Mainframe Modernization Ihre Daten mit einem AWS KMS Schlüssel, den AWS Mainframe Modernization besitzt und für Sie verwaltet. Sie können sich jedoch dafür entscheiden, einen vom Kunden verwalteten Schlüssel zu verwenden. AWS KMS

8. (Optional) Wählen Sie einen AWS KMS Schlüssel anhand des Namens oder des Amazon-Ressourcennamens (ARN) aus, oder wählen Sie `AWS KMS Schlüssel erstellen`, um zur AWS KMS Konsole zu wechseln und einen neuen AWS KMS Schlüssel zu erstellen.
9. (Optional) Wählen Sie im Abschnitt `Tags` die Option `Neues Tag hinzufügen` aus, um Ihrer Anwendung ein oder mehrere Anwendungs-Tags hinzuzufügen. Ein Anwendungs-Tag ist eine benutzerdefinierte Attributbezeichnung, mit der Sie Ihre AWS Ressourcen organisieren und verwalten können).
10. Wählen Sie `Weiter` aus.
11. Verwenden Sie im Abschnitt `Ressourcen und Konfigurationen` den `Inline-Editor`, um die Anwendungsdefinition einzugeben. Wählen Sie alternativ die Option `JSON-Datei für Anwendungsdefinitionen` in einem Amazon S3 S3-Bucket verwenden und geben Sie den Speicherort der Anwendungsdefinition an, die Sie verwenden möchten. Weitere Informationen finden Sie unter [AWS Beispiel für eine Blu-Age-Anwendungsdefinition](#) oder [Micro Focus-Anwendungsdefinition](#).
12. Wählen Sie `Weiter` aus.
13. Überprüfen Sie auf der Seite `Überprüfen und erstellen` die von Ihnen eingegebenen Informationen und wählen Sie `Anwendung erstellen`.

## Stellen Sie eine AWS Mainframe-Modernisierungsanwendung bereit

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine AWS Mainframe-Modernisierungsanwendung bereitzustellen.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

### Stellen Sie eine Anwendung bereit

Um eine AWS Mainframe-Modernisierungsanwendung auszuführen, müssen Sie sie zunächst in einer Laufzeitumgebung bereitstellen. Eine Anwendung kann mehr als eine Version haben. Jede Version einer Anwendung hat ihre eigene Anwendungsdefinition. Um eine Anwendung bereitzustellen, müssen Sie die Version angeben, die Sie bereitstellen möchten.

Sie können jeweils nur eine Version einer bestimmten Anwendung bereitstellen. Wenn Sie eine Version einer Anwendung bereitstellen und sich dann dafür entscheiden, stattdessen eine andere Version bereitzustellen, müssen Sie die Anwendung zunächst beenden, falls sie ausgeführt wird.

Um eine Anwendung bereitzustellen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Anwendung erstellen möchten.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, die Sie bereitstellen möchten.
4. Wählen Sie Anwendung bereitstellen aus.
5. Wählen Sie im Abschnitt Verfügbare Versionen die Version aus, die Sie bereitstellen möchten.
6. Wählen Sie im Abschnitt Umgebungen eine Laufzeitumgebung aus, in der Ihre Anwendung ausgeführt werden soll.
7. Wählen Sie Bereitstellen.

Um eine andere Version einer bereitgestellten Anwendung bereitzustellen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Anwendung erstellen möchten.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, die Sie bereitstellen möchten.
4. Wählen Sie im Menü Aktionen die Option Anwendung beenden.
5. Wählen Sie nach dem Beenden der Anwendung die Option Anwendung bereitstellen.
6. Wählen Sie im Abschnitt Verfügbare Versionen die Version aus, die Sie bereitstellen möchten. Im Abschnitt Umgebungen ist die Umgebung, in der die Anwendung bereits bereitgestellt ist, vorausgewählt.
7. Wählen Sie Bereitstellen.



# Aktualisieren Sie eine AWS Mainframe-Modernisierungsanwendung

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, um eine AWS Mainframe-Modernisierungsanwendung zu aktualisieren.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Aktualisieren einer Anwendung

Eine AWS Mainframe-Modernisierungsanwendung kann mehrere Versionen haben, von denen jede ihre eigene Anwendungsdefinition hat. Um eine Anwendung zu aktualisieren, geben Sie eine neue Anwendungsdefinition an. Dadurch wird eine neue Version der Anwendung erstellt.

Um eine Anwendung zu aktualisieren

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, die Sie aktualisieren möchten, erstellt wurde.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, die Sie aktualisieren möchten.
4. Wählen Sie auf der Seite mit den Anwendungsdetails im Abschnitt Aktuelle Definition die Option Bearbeiten aus, um die aktuelle Anwendungsdefinition zu aktualisieren.
5. Verwenden Sie auf der Seite „Anwendung aktualisieren“ den Inline-Editor, um die aktuelle Anwendungsdefinition zu aktualisieren.

Wählen Sie alternativ die Option JSON-Datei für Anwendungsdefinitionen in einem Amazon S3 S3-Bucket verwenden und geben Sie den Speicherort der Anwendungsdefinition an, die Sie verwenden möchten. Weitere Informationen finden Sie unter [AWS Beispiel für eine Blu-Age-Anwendungsdefinition](#) oder [Micro Focus-Anwendungsdefinition](#).

6. Wenn Sie mit der Aktualisierung der Anwendungsdefinition fertig sind, wählen Sie Aktualisieren.

**Note**

Nachdem Sie die Anwendung aktualisiert haben, müssen Sie sie erneut bereitstellen. Weitere Informationen finden Sie unter [Stellen Sie eine AWS Mainframe-Modernisierungsanwendung bereit](#).

## Löschen Sie eine AWS Mainframe-Modernisierungsanwendung aus einer Umgebung

Sie können eine AWS Mainframe-Modernisierungsanwendung mithilfe der AWS Mainframe-Modernisierungskonsole aus einer Umgebung löschen.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

### Löschen Sie eine Anwendung aus einer Umgebung

Wenn Sie eine AWS Mainframe-Modernisierungsanwendung löschen müssen und diese läuft, stellen Sie sicher, dass Sie sie zuerst beenden. Sie können den Status der Bewerbung auf der Seite „Anwendungen“ einsehen.

Um eine Anwendung aus einer Umgebung zu löschen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, die Sie aus der Umgebung löschen möchten, erstellt wurde.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, die Sie aus der Umgebung löschen möchten, und klicken Sie dann auf Aktionen.
4. (Optional) Wenn der Status der Anwendung lautet `Running`, wählen Sie Anwendung beenden.
5. Wählen Sie „Aus Umgebung löschen“.

Der Löschvorgang beginnt sofort.

# Löschen Sie eine AWS Mainframe-Modernisierungsanwendung

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, um eine AWS Mainframe-Modernisierungsanwendung zu löschen.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Löschen einer Anwendung

Wenn Sie eine AWS Mainframe-Modernisierungsanwendung löschen müssen und diese läuft, stellen Sie sicher, dass Sie sie zuerst beenden. Sie können den Status der Bewerbung auf der Seite „Anwendungen“ einsehen.

So löschen Sie eine Anwendung

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, die Sie löschen möchten, erstellt wurde.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, die Sie löschen möchten, und klicken Sie dann auf Aktionen.
4. (Optional) Wenn der Status der Anwendung lautet `Running`, wählen Sie Anwendung beenden.
5. Klicken Sie auf Delete Application (Anwendung löschen).
6. Geben Sie im Fenster „Anwendung löschen“ ein, `delete` um zu bestätigen, dass Sie die Anwendung löschen möchten, und wählen Sie dann Löschen.

## Reichen Sie Batch-Jobs für AWS Mainframe-Modernisierungsanwendungen ein

In AWS Mainframe Modernization können Sie Batch-Jobs für Ihre Anwendungen einreichen. Sie können Batch-Jobs einreichen oder stornieren und Details zu Batch-Job-Ausführungen überprüfen. Jedes Mal, wenn Sie einen Batch-Job einreichen, erstellt AWS Mainframe Modernization eine separate Batch-Job-Ausführung. Sie können die Ausführung dieses Jobs überwachen. Sie können anhand des Namens nach Batch-Jobs suchen und JCL- oder Skriptdateien für Batch-Jobs bereitstellen.

**⚠ Important**

Wenn Sie einen Batch-Job stornieren, wird der Job dadurch nicht gelöscht. Dadurch wird eine bestimmte Ausführung des Batch-Jobs abgebrochen. Die Batch-Job-Datensätze stehen Ihnen weiterhin in den Details zur Batch-Job-Ausführung zur Verfügung.

Wenn Ihr Batch-Job Zugriff auf einen oder mehrere Datensätze erfordert, verwenden Sie die AWS Mainframe-Modernisierungskonsole oder die AWS Command Line Interface (AWS CLI), um die Datensätze zu importieren. Weitere Informationen finden Sie unter [Importieren Sie Datensätze für AWS Mainframe-Modernisierungsanwendungen](#).

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) und in [Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS](#) abgeschlossen haben.

**Themen**

- [Reichen Sie einen Batch-Job ein](#)
- [Starten Sie einen Batch-Job neu](#)

## Reichen Sie einen Batch-Job ein

Um einen Batch-Job einzureichen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, für die Sie einen Batch-Job einreichen möchten, erstellt wurde.
3. Wählen Sie auf der Seite „Anwendungen“ die Anwendung aus, für die Sie einen Batch-Job einreichen möchten.

**ℹ Note**

Bevor Sie einen Batch-Job an eine Anwendung weiterleiten können, müssen Sie die Anwendung erfolgreich bereitstellen.

4. Wählen Sie auf der Seite mit den Anwendungsdetails die Option Batch-Jobs aus.

5. Wählen Sie Auftrag absenden.
6. Wählen Sie im Abschnitt „Skript auswählen“ ein Skript aus. Sie können anhand des Namens nach dem gewünschten Skript suchen.
7. Wählen Sie Auftrag absenden.

## Starten Sie einen Batch-Job neu

Um einen Batch-Job neu zu starten

### Important

Ein Neustart eines Batch-Jobs ist nur in den Micro Focus Environment Engine-Versionen 8.0.6 oder höher verfügbar. Außerdem muss ein EFS- oder FSx-Dateisystem an Ihre Umgebung angehängt sein.

1. [Öffnen Sie die AWS Mainframe Modernization Console unter https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung und Ihr Batch-Job erstellt wurden.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, in der Sie einen Batch-Job neu starten möchten.
4. Wählen Sie auf der Seite mit den Anwendungsdetails die Option Batch-Jobs aus.
5. Wählen Sie den Batch-Job, den Sie neu starten möchten, aus der generierten Liste aus. Navigieren Sie zum Menü Aktionen und wählen Sie Job neu starten.
6. Geben Sie an, wie Sie den Batch-Job neu starten möchten. Sie können wählen, ob Sie von Anfang an neu starten oder mit steps oder procsteps neu starten möchten.
  - Mit der Option Von Anfang neu starten können Sie alle Schritte eines Batchjobs von Anfang an neu starten.
  - Mit der Option Mit Schritten oder Procsteps neu starten können Sie einen bestimmten Schritt oder Procstep (Prozedurschritt) auswählen, den Sie neu starten möchten, und optional einen Schritt oder Procstep, nach dem Sie enden möchten.

**Note**

Der Endschrift oder Procstep muss größer oder gleich der Startschritt- oder Procstep-Nummer sein.

7. Wählen Sie Auftrag absenden.

## Importieren Sie Datensätze für AWS Mainframe-Modernisierungsanwendungen

Mit AWS Mainframe Modernization können Sie Datensätze importieren, um sie mit Ihren Anwendungen zu verwenden. Sie können die Datensätze in einer JSON-Datei angeben, die in einem Amazon S3 S3-Bucket gespeichert ist, oder Sie können die Konfigurationswerte für Datensätze separat angeben. Nachdem Sie die Datensätze importiert haben, können Sie die Details der Importaufgabe überprüfen, um sicherzustellen, dass die gewünschten Datensätze importiert wurden. Alle katalogisierten Datensätze für eine Anwendung werden zusammen in der Konsole aufgeführt.

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, um Datensätze für eine AWS Mainframe-Modernisierungsanwendung zu importieren.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) und in abgeschlossen haben. [Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS](#)

### Importieren Sie einen Datensatz

Um einen Datensatz zu importieren

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, für die Sie Datensätze importieren möchten, erstellt wurde.
3. Wählen Sie auf der Seite Anwendungen die Anwendung aus, für die Sie Datensätze importieren möchten.
4. Wählen Sie auf der Seite mit den Anwendungsdetails die Option Datensätze aus.

5. Wählen Sie Importieren aus.
6. Führen Sie eine der folgenden Aktionen aus:
  - Wählen Sie JSON-Datei zur Datensatzkonfiguration in einem Amazon S3 S3-Bucket verwenden und geben Sie den Speicherort der Datensatzkonfiguration an.
  - Wählen Sie Die Konfigurationswerte für den Datensatz separat mit geführter Konfiguration angeben aus. Spezifische Definitionsdetails finden Sie unter [the section called “Referenz zur Definition von Datensätzen”](#).

Geben Sie den Namen, die Datensatzorganisation (VSAM, GDG, PO, PS), den Standort und den externen Amazon S3 S3-Standort sowie die Parametereinstellungen für jeden Datensatz-Konfigurationswert ein. In der geführten Konfiguration können Sie auch JSON generieren wählen, um die JSON-Konfiguration anhand Ihrer Eingaben zu überprüfen.
7. Wählen Sie Absenden aus.

## Verwalten Sie Transaktionen für AWS Mainframe-Modernisierungsanwendungen

Mit AWS Mainframe Modernization können Sie eine Anwendung auf Anfrage gleichzeitig mit vielen anderen Benutzern ausführen, die Anfragen zur Ausführung derselben Anwendung mit denselben Dateien und Programmen einreichen. Eine einzelne Transaktion besteht aus einem oder mehreren Anwendungsprogrammen, die die erforderliche Verarbeitung durchführen.


Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) und in abgeschlossen haben [Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS](#).

### Transaktionen für Anwendungen verwalten

Um Transaktionen für Anwendungen zu verwalten


1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Anwendung, die Sie ausführen möchten, erstellt wurde.
3. Wählen Sie auf der Seite „Anwendungen“ die Anwendung aus, in der Sie Transaktionen verwalten möchten.

4. Wählen Sie auf der Registerkarte Transaktionen unter Transaktionsressourcen aus der Dropdownliste aus, wie Ihre Ressourcen angezeigt werden sollen. Sie können Ressourcen nach Transaktionsressourcen, Gruppen, Listen oder SITs anzeigen.
  - Mit Transaktionsressourcen können Sie den Ressourcentyp anhand von Dateidefinitionen, Transaktionsdefinitionen, Programmdefinitionen oder Definitionen für transiente Datenwarteschlangen auswählen.

 Note

Der AWS Mainframe Modernization Service unterstützt zusätzliche Ressourcentypen zur Verwaltung von Transaktionen für Anwendungen und kann über die Konsole aufgerufen werden.

- Gruppen sind eine Sammlung von Transaktionsressourcen. Sie können Gruppen auswählen, die Sie Ihrer Transaktionsressource zuordnen möchten.
- Listen sind eine geordnete Sammlung von Gruppen. Sie können alle Ihre Transaktionsressourcen und Gruppen in einer Listenansicht sehen. Die Startliste bestimmt, welche Ressourcen geladen werden, wenn der Server initialisiert wird.
  - Mit der AWS Blu Age Refactor Engine geben Sie die Listen an, die beim Start enthalten sein sollen. Die Anzahl der Listen ist unbegrenzt.
  - Mit der Micro Focus Replatform Engine können Sie bis zu vier Listen in einer SIT angeben.
- SIT (System Initialization Table) zeigt alle verfügbaren Transaktionskonfigurationen an. Sie können SITs anhand ihrer Eigenschaften (Name, Beschreibung und Startlisten) finden. Sie können auch Listen auswählen, die Sie mit Ihrer ausgewählten SIT verknüpfen möchten.

 Note

SITs gelten nur für die Micro Focus Replatform Engine.

5. Wählen Sie eine Transaktionsressource aus, um alle Ressourceninformationen anzuzeigen. Sie können auch alle Attribute anzeigen, die mit Ihrer Transaktionsressource verknüpft sind.



# AWS Ressourcen für eine migrierte Anwendung erstellen

Um Ihre migrierte Anwendung in ausführen zu können AWS, müssen Sie einige AWS Ressourcen zusammen mit anderen erstellen. AWS-Services Zu den Ressourcen, die Sie erstellen müssen, gehören:

- Ein S3-Bucket für Anwendungscode, Konfiguration, Datendateien und andere erforderliche Artefakte.
- Eine Amazon RDS- oder Amazon Aurora Aurora-Datenbank zur Speicherung der Daten, die die Anwendung benötigt.
- Eine AWS KMS key, die AWS Secrets Manager zum Erstellen und Speichern von Geheimnissen benötigt wird.
- Ein Secrets Manager Manager-Geheimnis zum Speichern der Datenbankanmeldedaten.

## Note

Jede migrierte Anwendung benötigt einen eigenen Satz dieser Ressourcen. Dies ist ein Mindestsatz. Ihre Anwendung benötigt möglicherweise auch zusätzliche Ressourcen, wie Amazon Cognito Cognito-Geheimnisse oder MQ-Warteschlangen.

## Erforderliche Berechtigungen

Stellen Sie sicher, dass Sie über die folgenden Berechtigungen verfügen:

- `s3:CreateBucket`, `s3:PutObject`
- `rds:CreateDBInstance`
- `kms:CreateKey`
- `secretsmanager:CreateSecret`

## Amazon-S3-Bucket

Sowohl für umgestaltete als auch für neu plattformierte Anwendungen ist ein S3-Bucket erforderlich, den Sie wie folgt konfigurieren:

```
bucket-name/root-folder-name/application-name
```

## bucket-name

Jeder Name, der den Beschränkungen der Amazon S3 S3-Benennung entspricht. Wir empfehlen, dass Sie den AWS-Region Namen als Teil Ihres Bucket-Namens angeben. Stellen Sie sicher, dass Sie den Bucket in derselben Region erstellen, in der Sie die migrierte Anwendung bereitstellen möchten.

## Name des Stammordners

Name, der erforderlich ist, um die Einschränkungen in der Anwendungsdefinition zu erfüllen, die Sie als Teil der Mainframe-Modernisierungsanwendung erstellen. AWS Sie können den verwenden `root-folder-name`, um zwischen verschiedenen Versionen einer Anwendung zu unterscheiden, z. B. V1 und V2.

## application-name

Der Name Ihrer migrierten Anwendung, zum Beispiel PlanetsDemo oder BankDemo.

# Datenbank

Sowohl für umgestaltete als auch für plattformübergreifende Anwendungen ist möglicherweise eine Datenbank erforderlich. Sie müssen die Datenbank gemäß den spezifischen Anforderungen für jede Runtime-Engine erstellen, konfigurieren und verwalten. AWS Die Mainframe-Modernisierung unterstützt die Verschlüsselung bei der Übertragung in dieser Datenbank. Wenn Sie SSL in Ihrer Datenbank aktivieren, stellen Sie sicher, dass Sie `sslMode` im Datenbankgeheimnis zusammen mit den Verbindungsdetails der Datenbank angeben. Weitere Informationen finden Sie unter [AWS Secrets Manager geheim](#).

Wenn Sie das AWS Blu Age-Refactoring-Pattern verwenden und eine BluSam Datenbank benötigen, erwartet die AWS Blu Age Runtime Engine eine Amazon Aurora PostgreSQL-Datenbank, die Sie erstellen, konfigurieren und verwalten müssen. Die BluSam Datenbank ist optional. Erstellen Sie diese Datenbank nur, wenn Ihre Anwendung sie benötigt. Um die Datenbank zu erstellen, folgen Sie den Schritten unter [Erstellen eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Wenn Sie das Micro Focus Replatforming Pattern verwenden, können Sie entweder eine Amazon RDS- oder eine Amazon Aurora PostgreSQL-Datenbank erstellen. Um die Datenbank zu erstellen, folgen Sie den Schritten unter [Erstellen einer Amazon RDS-DB-Instance](#) im Amazon RDS-

Benutzerhandbuch oder unter [Erstellen eines Amazon Aurora Aurora-DB-Clusters](#) im Amazon Aurora Aurora-Benutzerhandbuch.

Für beide Runtime-Engines müssen Sie die Datenbankanmeldedaten in AWS Secrets Manager Using an speichern AWS KMS key , um sie zu verschlüsseln.

## AWS Key Management Service Schlüssel

Sie müssen die Anmeldeinformationen für die Anwendungsdatenbank sicher in speichern AWS Secrets Manager. Um ein Geheimnis in Secrets Manager zu erstellen, müssen Sie ein erstellen AWS KMS key. Um einen KMS-Schlüssel zu erstellen, folgen Sie den Schritten [unter Schlüssel erstellen](#) im AWS Key Management Service Entwicklerhandbuch.

Nachdem Sie den Schlüssel erstellt haben, müssen Sie die Schlüsselrichtlinie aktualisieren, um AWS Mainframe Modernization Entschlüsselungsberechtigungen zu gewähren. Fügen Sie die folgenden Richtlinienerklärungen hinzu:

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

## AWS Secrets Manager geheim

Sie müssen die Anmeldeinformationen für die Anwendungsdatenbank sicher in speichern AWS Secrets Manager. Um einen geheimen Schlüssel zu erstellen, folgen Sie den Schritten [unter Erstellen eines geheimen AWS Secrets Manager Datenbankschlüssels](#) im Benutzerhandbuch.

AWS Mainframe Modernization unterstützt die Verschlüsselung bei der Übertragung in dieser Datenbank. Wenn Sie SSL in Ihrer Datenbank aktivieren, stellen Sie sicher, dass Sie `sslMode` im Datenbankgeheimnis zusammen mit den Verbindungsdetails der Datenbank angeben. Sie können einen der folgenden Werte für `sslMode`: `verify-full``verify-ca`, oder `disable`.

Wählen Sie während der Schlüsselerstellung die Option Ressourcenberechtigungen — optional und anschließend Berechtigungen bearbeiten aus. Fügen Sie im Richtlinieneditor eine ressourcenbasierte Richtlinie wie die folgende hinzu, um den Inhalt der verschlüsselten Felder abzurufen.

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

## Die verwaltete Anwendung konfigurieren

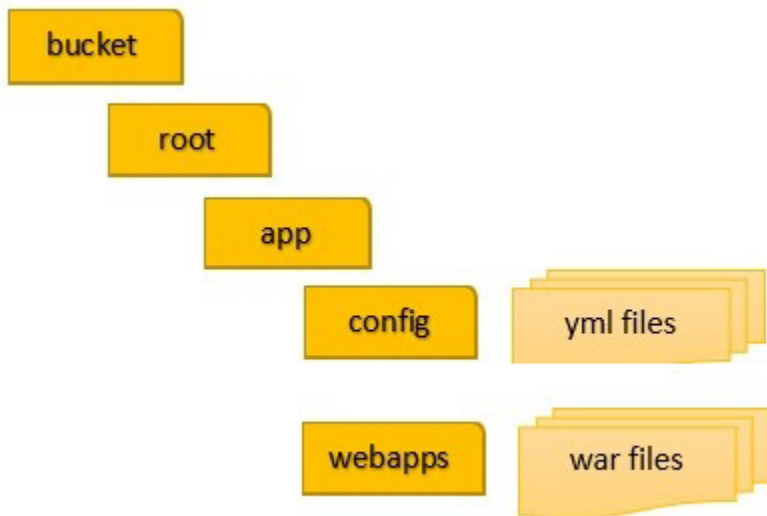
Sie können Ihre Anwendung so konfigurieren, dass sie Zugriff auf ältere Dienstprogramme bietet. Sie können auch zusätzliche Eigenschaften anpassen. Um zu verstehen, was Sie wo konfigurieren können, ist es hilfreich, die Gesamtstruktur einer modernisierten AWS Blu Age-Anwendung zu verstehen.

### Themen

- [Struktur der von AWS Blu Age verwalteten Anwendungen](#)
- [Konfiguration des Zugriffs auf Dienstprogramme für verwaltete Anwendungen](#)
- [Fügen Sie Konfigurationseigenschaften für die AWS Blu Age-Engine hinzu](#)

## Struktur der von AWS Blu Age verwalteten Anwendungen

Wenn Sie das AWS Blu Age-Refactoring-Pattern verwenden, erwartet die AWS Blu Age Runtime Engine die folgende Struktur innerhalb des `application-name` Ordners in Ihrem S3-Bucket:



### config

Enthält die YAML-Dateien für Ihr Projekt. Dies sind die für Ihre Anwendung spezifischen YAML-Dateien, die normalerweise so benannt sind, `application-planetsdemo.yaml` und nicht die `application-main.yaml` Datei, die AWS Mainframe Modernization automatisch für Sie bereitstellt und einrichtet.

### Webapps

Enthält die `war` Dateien für Ihre Anwendung. Diese Dateien sind ein Ergebnis des Modernisierungsprozesses.

Eine Anwendung kann auch die folgenden optionalen Ordner enthalten:

### ics/sql

Enthält das `initJics.sql` Skript, das die JICS-Datenbank für Ihre Anwendung initialisiert.

### Skripts

Enthält Anwendungsskripts, die Sie auch direkt in den Dateien bereitstellen können. `war`

### sql

Enthält Anwendungs-SQL-Dateien, die Sie auch direkt in den `war` Dateien bereitstellen können.

### Ink

Enthält LNK-Anwendungsdateien, die Sie auch direkt in den `war` Dateien bereitstellen können.

## extra

Enthält Jars, die zusätzliche Funktionen für die modernisierte Anwendung bieten können.

## Verwaltung des Java-Speicherverbrauchs einer Anwendung

Um den Java-Speicherverbrauch für die Anwendung zu verwalten, fügen Sie dem `application-name` Ordner eine Eigenschaftendatei mit dem Namen `tomcat.properties` hinzu. Diese Datei kann zwei Eigenschaften haben: `xms`, die den minimalen Java-Speicherverbrauch angibt, und `xmx`, die den maximalen Java-Speicherverbrauch angibt. Das Folgende ist ein Beispiel für den Inhalt einer gültigen `tomcat.properties` Datei.

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

Die Werte, die Sie für diese beiden Eigenschaften angeben, können in jeder der folgenden Einheiten angegeben werden:

- Byte: Geben Sie keine Einheit an.
- Kilobyte: Hängen Sie ein K an den Wert an.
- Megabyte: Hängen Sie ein M an den Wert an.
- Gigabyte: Hängen Sie ein G an den Wert an.

## Konfiguration des Zugriffs auf Dienstprogramme für verwaltete Anwendungen

Wenn Sie eine Mainframe-Anwendung mit AWS Blu Age umgestalten, müssen Sie möglicherweise Unterstützung für verschiedene ältere Plattform-Hilfsprogramme wie IDCAMS, INFUTILB, SORT usw. bereitstellen, falls Ihre Anwendung davon abhängt. AWS Das Blu Age-Refactoring ermöglicht diesen Zugriff mit einer speziellen Webanwendung, die zusammen mit modernisierten Anwendungen bereitgestellt wird. Für diese Webanwendung ist eine Konfigurationsdatei erforderlich `application-utility-pgm.yml`, die Sie bereitstellen müssen. Wenn Sie diese Konfigurationsdatei nicht bereitstellen, kann die Webanwendung nicht zusammen mit Ihrer Anwendung bereitgestellt werden und ist auch nicht verfügbar.

### Themen

- [Konfigurationseigenschaften](#)

In diesem Thema werden alle möglichen Eigenschaften, die Sie in der `application-utility-pgm.yml` Konfigurationsdatei angeben können, zusammen mit ihren Standardeinstellungen beschrieben. In diesem Thema werden sowohl erforderliche als auch optionale Eigenschaften beschrieben. Das folgende Beispiel ist eine vollständige Konfigurationsdatei. Sie listet die Eigenschaften in der von uns empfohlenen Reihenfolge auf. Sie können dieses Beispiel als Ausgangspunkt für Ihre eigene Konfigurationsdatei verwenden.

```
# 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
```

## Konfigurationseigenschaften

Sie können die folgenden Eigenschaften in Ihrer Konfigurationsdatei angeben.

### spring.jta.enabled

Optional. Steuert, ob die JTA-Unterstützung aktiviert ist. Für Dienstprogramme empfehlen wir, diesen Wert auf `false` festzulegen.

```
spring.jta.enabled : false
```

### logging.config

Erforderlich Gibt den Pfad zur dedizierten Logger-Konfigurationsdatei an. Wir empfehlen, den Namen zu verwenden `logback-utility.xml` und diese Datei als Teil der modernisierten Anwendung bereitzustellen. Die übliche Art, diese Dateien zu organisieren, besteht darin, alle Logger-Konfigurationsdateien am selben Ort abzulegen, normalerweise in dem Unterordner, in dem sich der Ordner `/config/logback` /`config` befindet, der die YAML-Konfigurationsdateien enthält. Weitere Informationen finden Sie in [Kapitel 3: Logback-Konfiguration in der Logback-Dokumentation](#).



```
logging.config : classpath:logback-utility.xml
```

## encoding

Erforderlich Gibt den Zeichensatz an, den das Hilfsprogramm verwendet. In den meisten Fällen, wenn Sie von z/OS-Plattformen migrieren, ist dieser Zeichensatz eine EBCDIC-Variante und sollte dem Zeichensatz entsprechen, der für die modernisierten Anwendungen konfiguriert ist. Die Standardeinstellung, falls nicht festgelegt, ist. ASCII

```
encoding : cp1047
```

## sys PunchEncoding

Optional. Gibt den Zeichensatz an, den INFUTILB und DSNUTILB zum Generieren und Lesen von SYSPUNCH-Dateien verwenden. Wenn Sie die SYSPUNCH-Dateien der Legacy-Plattform unverändert verwenden, sollte dieser Wert eine EBCDIC-Variante sein. Die Standardeinstellung, falls nicht gesetzt, ist. ASCII

```
sysPunchEncoding : cp1047
```

## Konfiguration der primären Datenquelle

Einige datenbankbezogene Dienstprogramme, wie LOAD und UNLOAD, erfordern den Zugriff auf eine Zieldatenbank über eine Datenquelle. Wie andere Datenquellendefinitionen innerhalb von AWS Mainframe Modernization erfordert dieser Zugriff, dass Sie Folgendes verwenden. AWS Secrets Manager Die Eigenschaften, die auf die richtigen Geheimnisse in Secrets Manager verweisen, lauten wie folgt:

```
spring.aws.client.datasources.primary.secret
```

Optional. Gibt das Geheimnis in Secrets Manager an, das die Datenquelleneigenschaften enthält.

```
spring.aws.client.datasources.primary.secret: datasource-secret-ARN
```

```
spring.aws.client.datasources.primary.dbname
```

Optional. Gibt den Namen der Zieldatenbank mit der Eigenschaft an, wenn der Datenbankname nicht direkt im geheimen Datenbankschlüssel angegeben ist. dbname

```
spring.aws.client.datasources.primary.dbname: target-database-name
```

## behandeln LargeNumber AsInteger

Optional. Bezieht sich auf die Besonderheiten der Oracle-Datenbank-Engine und die Verwendung der DSNTEP2/DSNTEP4-Dienstprogramme. Wenn Sie dieses Flag auf `true` setzen, werden große Zahlen aus der Oracle-Datenbank (NUMBER (38,0)) als Ganzzahlen behandelt. Standard: `false`

```
treatLargeNumberAsInteger : false
```

## ZonedMode

Optional. Legt den Zonenmodus fest, um in Zonen aufgeteilte Datentypen zu kodieren oder zu dekodieren. Diese Einstellung beeinflusst die Art und Weise, wie Zeichenziffern dargestellt werden. Die folgenden Werte sind gültig:

- **EBCDIC\_STRICT**: Standard. Verwenden Sie eine strikte Definition für die Behandlung von Zeichen. Je nachdem, ob es sich bei dem Zeichensatz um EBCDIC oder ASCII handelt, werden bei der Darstellung der Vorzeichen die folgenden Zeichen verwendet:
  - EBCDIC-Zeichen, die Byte (Cn+Dn) entsprechen und positive und negative Ziffernbereiche (bis, bis) darstellen. +0 +9 -0 -9 Die Zeichen werden als { , A bis I, } bis angezeigt J R
  - ASCII-Zeichen, die Byte (3n+7n) entsprechen und positive und negative Ziffernbereiche (bis+9, +0 -0 bis-9) darstellen. Die Zeichen werden als „bis“, „0bis9“ angezeigt p y
- **EBCDIC\_MODIFIED**: Verwenden Sie eine modifizierte Definition für die Behandlung von Zeichen. Sowohl bei EBDIC als auch bei ASCII steht dieselbe Zeichenliste für die Ziffern der Vorzeichen, d. h. für die Zuordnung zu + zu und für die +9 Zuordnung +0 zu { + A zu. I -0 -9 } J R \
- **AS400**: Wird für modernisierte Altgeräte verwendet, die von iSeries-Plattformen (AS400) stammen.

```
zonedMode : EBCDIC_STRICT
```

## jcl.type

Optional. Zeigt den älteren Typ modernisierter JCL-Skripten an. Das IDCAMS-Hilfsprogramm verwendet diese Einstellung, um den Rückgabecode anzupassen, wenn es sich bei der aufrufenden JCL um einen Typ handelt. v se Gültige Werte sind:

- mvs (Standard)
- vse

```
jcl.type : mvs
```

## Verwandte Eigenschaften der Dienstprogramme zum Entladen von Datenbanken

Verwenden Sie diese Eigenschaften, um Dienstprogramme zu konfigurieren, die Datenbanktabellen in Datensätze entladen. Alle der folgenden Eigenschaften sind optional.

Dieses Beispiel zeigt alle möglichen Entladeeigenschaften.

```
# 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
```

## SQL Shift CodePoint

Optional. Gibt einen Ganzzahlwert an, der die für Daten verwendete SQL-Codepunktverschiebung darstellt. Der Standardwert ist 0. Das bedeutet, dass keine Codepunktverschiebung vorgenommen wird. Passen Sie diese Einstellung an den SQL-Codepunkt-Shift-Parameter an, der für modernisierte Anwendungen verwendet wird. Wenn Codepunkt-Shifting verwendet wird, ist der häufigste Wert für diesen Parameter 384.

```
unload.sqlCodePointShift: 0
```

## nbi

Optional. Gibt ein Null-Indikator-Byte an. Dies ist ein Hexadezimalwert (als Zeichenfolge), der rechts neben dem Datenwert hinzugefügt wird. Die beiden möglichen Werte lauten wie folgt:

- **WhenNull:** Fügt den Hexadezimalwert hinzu, wenn der Datenwert Null ist. Der Standardwert ist `6``. Manchmal wird stattdessen der hohe Wert `FF` verwendet.

```
unload.nbi.whenNull: "6F"
```

- **wann NotNull:** Fügt den Hexadezimalwert hinzu, wenn der Datenwert nicht Null ist, die Spalte aber nullwertfähig ist. Die Standardeinstellung ist `00` (niedriger Wert).

```
unload.nbi.whenNotNull: "00"
```

## verwenden DatabaseConfiguration

Optional. Gibt die Formatierungseigenschaften für Datum und Uhrzeit an. Dies wird verwendet, um mit Datums-/Uhrzeitobjekten in UNLOAD-Abfragen umzugehen. Der Standardwert ist `false`.

- Wenn auf `gesetztrue`, werden die `pgmTimestampFormat` Eigenschaften `pgmDateFormat``pgmTimeFormat`, und aus der Hauptkonfigurationsdatei (`application-main.yml`) verwendet.
- Wenn auf `gesetzfalse`, werden die folgenden Formatierungseigenschaften für Datum und Uhrzeit verwendet:
  - `unload.format.date:` Gibt ein Formatierungsmuster für das Datum an. Der Standardwert ist `MM/dd/yyyy`.
  - `unload.format.time:` Gibt ein Formatierungsmuster für die Uhrzeit an. Der Standardwert ist `HH.mm.ss`.
  - `unload.format.timestamp:` Gibt ein Zeitstempel-Formatierungsmuster an. Der Standardwert ist `yyyy-MM-dd-HH.mm.ss.SSSSSS`.

## Chunk-Größe

Optional. Gibt die Größe der Datenblöcke an, die zur Erstellung von SYSREC-Datensätzen verwendet werden. Diese Datensätze sind das Ziel des Datensatz-Entladevorgangs mit parallel Vorgängen. Die Standardeinstellung ist `0` (keine Blöcke).

```
unload.chunkSize:0
```

## Größe abrufen

Optional. Gibt die Größe des Datenabrufs an. Der Wert ist die Anzahl der Datensätze, die gleichzeitig abgerufen werden sollen, wenn eine Datenblockstrategie verwendet wird. Standard: 0.

```
unload.fetchSize:0
```

## var Null Charls

Optional. Gibt an, wie mit einer Varchar-Spalte mit leerem Inhalt umgegangen werden soll, die keine NULL-Werte zulassen. Der Standardwert ist `false`.

Wenn Sie diesen Wert auf `true` setzen, wird der Spalteninhalt für Entladezwecke als leere Zeichenfolge und nicht als einzelne Leerzeichenfolge behandelt. Setzen Sie dieses Flag nur `true` für den Fall der Oracle-Datenbank-Engine auf.

```
unload.varCharIsNull: false
```

## ColumnFiller

Optional. Gibt den Wert an, der zum Auffüllen entladener Spalten in Varchar-Spalten verwendet werden soll. Mögliche Werte sind Leerzeichen oder niedrige Werte. Die Standardeinstellung ist Leerzeichen.

```
unload.columnFiller: space
```

## Eigenschaften im Zusammenhang mit dem Laden der Datenbank

Verwenden Sie diese Eigenschaften, um Dienstprogramme zu konfigurieren, die Datensatzdatensätze in eine Zieldatenbank laden, z. B. DSNUTILB. Alle folgenden Eigenschaften sind optional.

Dieses Beispiel zeigt alle möglichen Ladeeigenschaften.

```
# 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
```

## SQL CodePoint Shift

Optional. Gibt einen Ganzzahlwert an, der die SQL-Codepunktverschiebung darstellt, die für Daten verwendet wird. Der Standardwert ist 0, was bedeutet, dass Anwendungen keine Codepunktverschiebung vornehmen. Passen Sie diese Einstellung an den SQL-Codepunkt-Shift-Parameter an, der für modernisierte Anwendungen verwendet wird. Wenn Sie Codepunktverschiebungen verwenden, ist der häufigste Wert für diesen Parameter 384.

```
load.sqlCodePointShift : 384
```

## batchSize

Optional. Gibt einen Ganzzahlwert an, der die Anzahl der Datensätze darstellt, die behandelt werden sollen, bevor Sie eine tatsächliche Batch-Anweisung an die Datenbank senden. Der Standardwert ist 0.

```
load.batchSize: 500
```

## Format

Optional. Gibt die Formatierungsmuster für Datum und Uhrzeit an, die für Datums-/Uhrzeitkonvertierungen beim Laden der Datenbank verwendet werden sollen.

- `load.format.localDate`: Lokales Formatierungsmuster für das Datum. Der Standardwert ist `dd.MM.yyyy|dd/MM/yyyy|yyyy-MM-dd`.
- `load.format.dbDate`: Formatierungsmuster für das Datenbankdatum. Der Standardwert ist `yyyy-MM-dd`.
- `load.format.localTime`: Formatierungsmuster für die Ortszeit. Der Standardwert ist `HH:mm:ss|HH.mm.ss`.

- `load.format.dbTime`: Formatierungsmuster für die Datenbankzeit. Der Standardwert ist `HH:mm:ss`.

## Tabellenzuordnungen

Optional. Gibt eine Sammlung von vom Kunden bereitgestellten Zuordnungen zwischen alten und modernen Tabellennamen an. Das Hilfsprogramm DSNUTILB verwendet diese Zuordnungen.

Geben Sie die Werte im folgenden Format an: MODERN\_TABLE\_NAME:  
LEGACY\_TABLE\_NAME

Ein Beispiel:

```
table-mappings:  
TABLE_1_NAME : LEGACY_TABLE_1_NAME  
TABLE_2_NAME : LEGACY_TABLE_2_NAME  
...  
TABLE_*N*_NAME : LEGACY_TABLE_*N*_NAME
```

### Note

Wenn die Utility-Anwendung gestartet wird, protokolliert sie explizit alle bereitgestellten Zuordnungen.

## Fügen Sie Konfigurationseigenschaften für die AWS Blu Age-Engine hinzu

Sie können dem `config` Ordner für Ihre umgestaltete Anwendung eine Datei hinzufügen, mit der Sie auf neue Funktionen in der Blu Age Runtime Engine zugreifen können. AWS Sie müssen dieser Datei einen Namen geben. `user-properties.yml` Diese Datei ersetzt die Anwendungsdefinition nicht, sondern erweitert sie. In diesem Thema werden die Eigenschaften beschrieben, die Sie in die `user-properties.yml` Datei aufnehmen können.

### Note

Sie können einige Parameter nicht ändern, da sie entweder von der AWS Mainframe-Modernisierung oder von der Anwendungsdefinition gesteuert werden. Alle in der Anwendungsdefinition für Ihre Anwendung definierten Parameter haben Vorrang vor den Parametern, die Sie in `user-properties.yml` angeben.

Weitere Hinweise zur Struktur umgestalteter Anwendungen finden Sie unter. [Struktur der von AWS Blu Age verwalteten Anwendungen](#)

Das folgende Diagramm zeigt, wo sich die `user-properties.yml` Datei innerhalb der Struktur der AWS Blu Age-Beispielanwendung befindet. PlanetsDemo

```
PlanetsDemo-v1/  
  ## config/  
  # ## application-PlanetsDemo.yml  
  # ## user-properties.yml  
  ## jics/  
  ## webapps/
```

## Referenz der Konfigurationseigenschaften

Dies ist die Liste der verfügbaren Eigenschaften. Alle Parameter sind optional.

### Themen

- [Eigenschaften der Gapwalk-Anwendung](#)
- [Eigenschaften des Gapwalk-Batchskripts](#)
- [Eigenschaften von Gapwalk Blugen](#)
- [Eigenschaften des Gapwalk CL-Befehls](#)
- [Eigenschaften des Gapwalk CL-Runners](#)
- [Gapwalk JHDB-Eigenschaften](#)
- [Gapwalk JICS-Eigenschaften](#)
- [Eigenschaften der Gapwalk-Laufzeit](#)
- [Eigenschaften des Gapwalk-Hilfsprogramms](#)
- [Andere Eigenschaften](#)

### Eigenschaften der Gapwalk-Anwendung

#### Bluesam.FileLoading.CommitInterval

Optional. Das BluSAM-Commit-Intervall.

Typ: Zahl

Standard: 100000



## card.encoding

Optional. Kartenkodierung: zur Verwendung mit. `useControlMVariable`

Typ: Zeichenfolge

Standard: CP1145

Überprüfen Sie die Größe der Eingabedatei

Optional. Gibt an, ob eine Prüfung ausgelöst werden soll, wenn die Dateigröße ein Vielfaches der Datensatzgröße ist.

Typ: Boolesch

Standard: false

## database.cursor.overflow.allowed

Optional. Gibt an, ob der Cursorüberlauf zugelassen werden soll. Wird auf `gesetzttrue`, um den nächsten Aufruf des Cursors unabhängig von seiner Position auszuführen. Legt fest, ob überprüft werden `false` soll, ob sich der Cursor an der letzten Position befindet, bevor ein nächster Cursor aufgerufen wird. Nur aktivieren, wenn der Cursor SCROLLABLE (SENSITIVE oder INSENSITIVE) ist

Typ: Boolesch

Standard: true

## InvalidNumericDataSimplifier.ON Daten

Optional. Wie man reagiert, wenn ungültige numerische Daten dekodiert werden. Zulässige Werte sind `reject,toleratespaces,toleratespaceslowvalues,toleratemost`.

Typ: Zeichenfolge

Standard: Ablehnen

## KeepExistingStandarddateien

Optional. Gibt an, ob der vorherige Standardwert des Datensatzes festgelegt werden soll.

Typ: Boolesch

Standard: false

## disposition.checkexistence

Optional. Gibt an, ob eine Überprüfung der Existenz einer Datei für Dataset mit DISP SHR oder OLD freigegeben werden soll.

Typ: Boolesch

Standard: false

## ExternalSort.Threshold

Optional. Der Schwellenwert für das Sortieren: wann zur externen Sortierung (Zusammenführung) gewechselt werden soll.

Typ: Zeichenfolge

Standard: null

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

## ForceHR

Optional. Gibt an, ob menschenlesbares SYSPRINT entweder für die Konsolen- oder Dateiausgabe verwendet werden soll.

Typ: Boolesch

Standard: false

## ForcedDate

Optional. Erzwingt ein bestimmtes Datum und eine bestimmte Uhrzeit in der Datenbank. Nur während der Entwicklung und beim Testen verwenden.

Standard: null

```
forcedDate: 2022-08-26T12:59:58.123456+01:57
```

## FrozenDate

Optional. Friert Datum und Uhrzeit in der Datenbank ein. Nur während der Entwicklung und beim Testen verwenden.

Standard: false

```
frozenDate: false
```

## ims.messages.ExtendedSize

Optional. Gibt an, ob die ExtendedSize für IMS-Nachrichten festgelegt werden soll.

Typ: Boolesch

Standard: false

## lockTimeout

Optional. Das Timeout in Millisekunden einer Transaktion, wenn innerhalb eines bestimmten Zeitraums keine Sperre aktiviert werden kann.

Typ: Zahl

Standard: 500

## MapTransfo.Präfixe

Optional. Liste der Präfixe, die bei der Transformation von ControlM-Variablen verwendet werden sollen. Jedes ist durch ein Komma getrennt.

Typ: Zeichenfolge

Standard: &, @, %%

## query.use ConcatCondition

Optional. Gibt an, ob die Schlüsselbedingung durch Schlüsselverkettung erstellt wird oder nicht.

Typ: Boolesch

Standard: false

## Rollback auf RTE

Optional. Gibt an, ob eine implizite Run-Unit-Transaktion bei Laufzeitausnahmen rückgängig gemacht werden soll.

Typ: Boolesch

Standard: false

## sct ThreadLimit

Optional. Das Thread-Limit für das Auslösen von Skripten.

Typ: Zahl

Standard: 5

#### SQL Shift CodePoint

Optional. Die Verschiebung des SQL-Codepunkts. Verschiebt den Codepunkt für Steuerzeichen, auf die wir bei der Migration älterer RDBMS-Daten auf moderne RDBMS-Daten stoßen könnten. Sie könnten beispielsweise angeben, dass das Unicode-Zeichen entspricht. `384 \u0180`

Typ: Zahl

Standard: 0

#### sql IntegerOverflow Zulässig

Optional. Gibt an, ob der SQL-Ganzzahlüberlauf zulässig ist, d. h. ob das Platzieren größerer Werte in der Hostvariablen zulässig ist.

Typ: Boolesch

Standard: false

#### Schritt: Abend FailWhen

Optional. Gibt an, ob ein Abend ausgelöst werden soll, wenn ein Schritt fehlschlägt oder die Ausführung abschließt.

Typ: Boolesch

Standard: true

#### Stopp ExecutionWhen ProgNot gefunden

Optional. Gibt an, ob die Ausführung beendet werden soll, wenn ein Programm nicht gefunden wird. Wenn auf `gesetzt true`, wird die Ausführung unterbrochen, wenn kein Programm gefunden wird.

Typ: Boolesch

Standard: true

#### Großbuchstaben userInput

Optional. Gibt an, ob Benutzereingaben in Großbuchstaben erfolgen müssen.

Typ: Boolesch

Standard: true

Verwenden Sie die Variable ControlM

Optional. Gibt an, ob die Control-M-Spezifikation für die Variablenersetzung verwendet werden soll.

Typ: Boolesch

Standard: false

Eigenschaften des Gapwalk-Batchskripts

encoding

Optional. Die in Batchscript-Projekten verwendete Kodierung (nicht bei Groovy). Erwartet eine gültige Kodierung CP1047,,,... IBM930 ASCII UTF-8

Typ: Zeichenfolge

Standard: ASCII

Eigenschaften von Gapwalk Blugen

managers.trancode

Optional. Der Dialogmanager Trancode-Mapping. Ermöglicht die Zuordnung eines JICS-Transaktionscodes zu einem Dialogmanager. Das erwartete Format ist `trancode1:dialogManager1;trancode2:dialogManager2;`

Typ: Zeichenfolge

Standard: null

`managers.trancode: 0R12:MYDIALOG1`

Eigenschaften des Gapwalk CL-Befehls

Befehle aus

Optional. Liste der Befehle, die ausgeschaltet werden sollen, getrennt durch Kommas. Zulässige Werte sind

PGM\_BASICRCVMSG,SNDRCVF,CHGVAR,QCLRDTAQ,RTVJOB,ADDLFM,ADDPFM,,RCVF,OVRDBF,DLTOVRCPYF,SNDDTAQ. Nützlich, wenn Sie ein vorhandenes Programm deaktivieren oder überschreiben möchten. PGM\_BASIC ist ein spezielles AWS Blu Age Runtime-Programm, das für Debugging-Zwecke entwickelt wurde.

Typ: Zeichenfolge

Standard: null

spring.datasource.primary.jndi-name

Optional. Die primäre JNDI-Datenquelle (Java Naming And Directory Interface).

Typ: Zeichenfolge

Standard: jdbc/primary

Zonierter Modus

Optional. Der Modus für die Kodierung oder Dekodierung von in Zonen aufgeteilten Datentypen. Zulässige Werte sind EBCDIC\_STRICT//EBCDIC\_MODIFIED. AS400

Typ: Zeichenfolge

Standard: EBCDIC\_STRICT

Eigenschaften des Gapwalk CL-Runners

cl.configuration.context.encoding

Optional. Die Kodierung von CL-Dateien. Erwartet eine gültige Kodierung CP1047IBM930,,ASCII,UTF-8...

Typ: Zeichenfolge

Standard: CP297

cl.ZonedMode

Optional. Der Modus für die Kodierung oder Dekodierung von CL-Befehlen (Control Language). Zulässige Werte sind EBCDIC\_STRICT/EBCDIC\_MODIFIED/AS400.

Typ: Zeichenfolge

Standard: EBCDIC\_STRICT

## Gapwalk JHDB-Eigenschaften

### ims.programs

Optional. Liste der zu verwendenden IMS-Programme. Trennen Sie jeden Parameter durch ein Semikolon (;) und jede Transaktion durch ein Komma (,). , Zum Beispiel: `ims.programs: PCP008, PCT008; PCP054, PCT054; PCP066, PCT066; PCP068, PCT068;`

Typ: Zeichenfolge

Standard: null

### jhdb.CheckPointPath

Optional. `jhdb.checkpointPersistencelst none` dies nicht der Fall, können Sie mit diesem Parameter den Checkpoint-Persistenzpfad (Speicherort der Datei `checkpoint.dat`) einrichten. Alle in der Registrierung enthaltenen Checkpoint-Daten werden serialisiert und in einer Datei (`checkpoint.dat`) gesichert, die sich im angegebenen Ordner befindet. Beachten Sie, dass nur Checkpoint-Daten (ScriptID, StepID, Datenbankposition und Checkpoint-Bereich) von dieser Sicherung betroffen sind.

Typ: Zeichenfolge

Standard: Datei: `./setup/`

### jhdb.CheckPointPersistenz

Optional. Der Checkpoint-Persistenzmodus. Zulässige Werte sind `none//add. end` Wird verwendet `add`, um Checkpoints beizubehalten, wenn ein neuer erstellt und der Registrierung hinzugefügt wird. Wird verwendet `end`, um Checkpoints beim Herunterfahren des Servers beizubehalten. Alle anderen Werte deaktivieren die Persistenz. Beachten Sie, dass jedes Mal, wenn ein neuer Checkpoint zur Registrierung hinzugefügt wird, alle vorhandenen Checkpoints serialisiert und die Datei gelöscht wird. Es handelt sich nicht um eine Anlage an die vorhandenen Daten in der Datei. Je nach Anzahl der Checkpoints kann dies also Auswirkungen auf die Leistung haben.

Typ: Zeichenfolge

Standard: keiner

## jhdb.configuration.context.encoding

Optional. Die JHDB-Kodierung (Java Hierarchical Database). Erwartet eine gültige Kodierungszeichenfolge CP1047,,,... IBM930 ASCII UTF-8

Typ: Zeichenfolge

Standard: CP297

## jhdb.identification.CardData

Optional. Wird verwendet, um einige „Personalausweisdaten“ fest in das durch den CARD-Parameter angegebene MID-Feld zu codieren.

Typ: Zeichenfolge

Standard: ""

## jhdb.lterm

Optional. Ermöglicht es Ihnen, bei einer IMS-Emulation eine gemeinsame logische Terminal-ID zu erzwingen. Wenn nicht gesetzt, wird sessionId verwendet.

Typ: Zeichenfolge

Standard: null

## jhdb.metadata.extrapath

Ein Konfigurationsparameter, der einen zusätzlichen, laufzeitspezifischen Stammordner für psbs- und dbds-Ordner angibt.

Typ: Zeichenfolge

Standard: Datei: ./setup/

### Note

Derzeit müssen Sie aufgrund von Einschränkungen bei der Bereitstellung Ihre Verzeichnisse dbds und psbs in das Konfigurationsverzeichnis Ihrer Anwendung oder in ein Unterverzeichnis des Konfigurationsverzeichnisses kopieren: z. B. config/setup

```
config
```



```
| - setup  
  | - dbds  
  | - psbs
```

und legen Sie es in application-jhdb.yml fest

```
jhdb.metadata.extrapath: file: ./config/setup/
```

### jhdb.navigation.cachenexts

Optional. Die Cache-Dauer (in Millisekunden), die bei der hierarchischen Navigation für ein RDBMS verwendet wird.

Typ: Zahl

Standard: 5000

### jhdb.query.limit JoinUsage

Optional. Gibt an, ob der Parameter Limit für die Verwendung von Joins in RDBMS-Diagrammen verwendet werden soll.

Typ: Boolesch

Standard: true

### jhdb.use-db-Präfix

Optional. Gibt an, ob ein Datenbankpräfix in der hierarchischen Navigation für ein RDBMS aktiviert werden soll.

Typ: Boolesch

Standard: true

### Gapwalk JICS-Eigenschaften

#### Standort von JsonInit jics.data.data

Optional. Speicherort der JSON-Datei, die vom Analyzer anhand der CSD-Analyse erstellt und zur Initialisierung der JICS-Datenbank verwendet wurde,

Typ: Zeichenfolge

Standard: ""

#### jics.db.data ScriptLocation

Optional. Speicherort des Skripts `initJics.sql`, das von Analyzer anhand der Analyse von CSD-Exporten vom Mainframe erstellt wurde.

Typ: Zeichenfolge

Standard: ""

#### Speicherort TestQuery von jics.db.data

Optional. Speicherort eines SQL-Skripts, das eine einzelne SQL-Abfrage enthält, von der erwartet wird, dass sie eine Anzahl von Objekten zurückgibt (z. B. das Zählen der Datensätze in der JICS-Programmtabelle). Wenn die Anzahl gleich 0 ist, wird die Datenbank mit dem `jics.db.dataScriptLocation` Skript geladen, andernfalls wird das Laden der Datenbank übersprungen.

Typ: Zeichenfolge

Standard: ""

#### jics.db.ddl ScriptLocation

Optional. Der Speicherort des Jics-DDL-Skripts. Ermöglicht es Ihnen, das Jics-Datenbankschema mithilfe eines SQL-Skripts zu initiieren.

Typ: Zeichenfolge

Standard: ""

```
jics.db.ddlScriptLocation: ./jics/sql/jics.sql
```

#### jics.db.schema-Speicherort TestQuery

Optional. Speicherort der SQL-Datei, die eine eindeutige Abfrage enthalten soll, die die Anzahl der Objekte im Jics-Schema (falls vorhanden) zurückgibt.

Typ: Zeichenfolge

Standard: ""

## jics.run.Pool.enable.UnitLauncher

Optional. Gibt an, ob der Run Unit Launcher-Pool in JICS aktiviert werden soll.

Typ: Boolesch

Standard: false

## jics.run.UnitLauncher.Pool.Size

Optional. Die Größe des Run-Unit-Launcher-Pools in JICS.

Typ: Zahl

Standard: 20

## UnitLauncher.jics.run.pool.ValidationInterval

Optional: Das Validierungsintervall des Run Unit Launcher-Pools in JICS, ausgedrückt in Millisekunden.

Typ: Zahl

Standard: 1000

## jics.queues.sqs.region

Optional. Das AWS-Region für Amazon SQS, das in JICS verwendet wird. Aus Leistungsgründen wird empfohlen, dieselbe Region für die bereitgestellte Anwendung festzulegen, dies ist jedoch nicht zwingend erforderlich.

Typ: Zeichenfolge

Standard: eu-west-1

## jics.xa.agent.timeout

Optional. Definiert die maximale Dauer, die der xa-Agent, der für die Verwaltung verteilter Transaktionen verantwortlich ist, benötigt, um seine Operationen abzuschließen.

Typ: Zahl

Standard: null

## mq.queues.sqs.region

Optional. Der AWS-Region für den Amazon SQS MQ-Service.

Typ: Zeichenfolge

Standard: eu-west-3

## TaskExecutor.Allow CoreThread TimeOut

Optional. Gibt an, ob ein Timeout für Kern-Threads in JCIS zulässig ist. Dies ermöglicht dynamisches Wachsen und Verkleinern auch in Kombination mit einer Warteschlange, die nicht Null ist (da die maximale Poolgröße erst zunimmt, wenn die Warteschlange voll ist).

Typ: Boolesch

Standard: false

## TaskExecutor.Core PoolSize

Optional. Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter, um die Größe des Core-Pools einzurichten.

Typ: Zahl

Standard: 5

## TaskExecutor.max PoolSize

Optional. Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter, um die maximale Poolgröße (maximale Anzahl parallel Threads) einzurichten.

Typ: Zahl

Standard: 10

## TaskExecutor.QueueCapacity

Optional. Wenn eine Transaktion in einem Terminal über ein Groovy-Skript initiiert wird, wird ein neuer Thread erstellt. Verwenden Sie diesen Parameter, um die Warteschlangengröße einzurichten. (= maximale Anzahl ausstehender Transaktionen, wenn sie `taskExecutor.maxPoolSize` erreicht ist)

Typ: Zahl

Standard: 50

## Eigenschaften der Gapwalk-Laufzeit

### Metadaten zwischenspeichern

Optional. Gibt an, ob Datenbank-Metadaten zwischengespeichert werden sollen.

Typ: Boolesch

Standard: true

### check-groovy-file

Optional. Gibt an, ob der Inhalt von Groovy-Dateien vor der Registrierung überprüft werden soll.

Typ: Boolesch

Standard: true

### Datenbank-Statistiken

Optional. Gibt an, ob SQL-Builder Statistikinformationen sammeln und anzeigen dürfen.

Typ: Boolesch

Standard: false

### Datum TimeFormat

Optional. Das `dateTimeFormat` beschreibt, wie der Datums- und Zeitstempeltyp einer Datenbank in Data Simplifier-Entitäten eingegeben wird. Zulässige Werte sind `///ISOEURUSALOCAL`

Typ: Zeichenfolge

Standard: ISO

### db DateFormat

Optional. Das Zieldatumsformat der Datenbank.

Typ: Zeichenfolge

Standard: yyyy-MM-DD

#### db TimeFormat

Optional. Das Zielzeitformat der Datenbank.

Typ: Zeichenfolge

Standard: hh:mm:ss

#### db TimestampFormat

Optional. Das Zeitstempelformat für das Datenbankziel.

Typ: Zeichenfolge

Standard: yyyy-MM-DD hh:mm:ss.ssssss

#### Größe abrufen

Optional. Der FetchSize-Wert für Cursor. Wird verwendet, wenn Daten mithilfe von Chunks mithilfe von Load/Unload-Utills abgerufen werden.

Typ: Zahl

Standard: 10

#### TrimStringForceDeaktiviert den SQL-Typ

Optional. Gibt an, ob das Kürzen aller SQL-Zeichenkettenparameter deaktiviert werden soll.

Typ: Boolesch

Standard: false

#### lokal DateFormat

Optional. Liste der lokalen Datumsformate. Trennen Sie jedes Format durch | .

Typ: Zeichenfolge

#### lokal TimeFormat

Optional. Liste der lokalen Zeitformate. Trennen Sie jedes Format durch | .

Typ: Zeichenfolge

## lokal TimestampFormat

Optional. Liste der lokalen Zeitstempelformate. Trennen Sie jedes Format durch | .

Typ: Zeichenfolge

Standard:

## pgm DateFormat

Optional. Das in den Programmen verwendete Datums- und Uhrzeitformat.

Typ: Zeichenfolge

Standard: yyyy-MM-DD

## pgm TimeFormat

Optional. Das Zeitformat, das für die Ausführung von PGM (Programmen) verwendet wird.

Typ: Zeichenfolge

Standard: HH.MM.ss

## pgm TimestampFormat

Optional. Das Zeitstempelformat.

Typ: Zeichenfolge

Standard: yyyy-MM-dd-hh.mm.ss.ssssssss

## Eigenschaften des Gapwalk-Hilfsprogramms

### jcl.type

Fakultativ. . jclDateityp. Zulässige Werte sind jcl/vse. Die PRINT/REPRO-Befehle des IDCAMS-Hilfsprogramms geben 4 zurück, wenn die Datei für JCL-Dateien, die nicht mit VSE verknüpft sind, leer ist.

Typ: Zeichenfolge

Standard: mvs

### listcat.variablelengthpreprocessor.enabled

Optional. Gibt an, ob der Präprozessor mit variabler Länge für den LISTCAT-Befehl aktiviert werden soll.

Typ: Boolesch

Standard: false

### listcat.variablelengthpreprocessor.type

Optional. Der Typ der Objekte, die in der Listcat-Datei enthalten sind, sofern Sie sie aktivieren. `listcat.variablelengthpreprocessor.enabled` Zulässige Werte sind `rdw/bdw`.

Typ: Zeichenfolge

Standard: `rdw`

### load.BatchSize

Optional. Die Batchgröße des Ladedienstprogramms.

Typ: Zahl

Standard: 0

### load.format.dbDate

Optional. Das zu verwendende Datenbankformat des Load Utility.

Typ: Zeichenfolge

Standard: `yyyy-MM-DD`

### load.format.DBTime

Optional. Die zu verwendende Uhrzeit der Load Utility-Datenbank.

Typ: Zeichenfolge

Standard: `hh:mm:ss`

### load.format.LocalDate

Optional. Das lokale Datumsformat des Ladedienstprogramms, das verwendet werden soll.



Typ: Zeichenfolge

Standard: dd.mm.yyyy|dd/MM/YYYY|yyyy-MM-DD

load.format.LocalTime

Optional. Das zu verwendende lokale Zeitformat des Ladedienstprogramms.

Typ: Zeichenfolge

Standard: hh:mm:ss|hh.mm.ss

CodePointload.sql Shift

Optional. Das SQL-Code-Hilfsprogramm Pointshift for Load. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank von DB2 Postgresql ist.

Typ: Zahl

Standard: 0

sys PunchEncoding

Optional. Der Zeichensatz für die Syspunch-Kodierung. Unterstützte Werte sind/Cp1047. ASCII

Typ: Zeichenfolge

Standard: ASCII

behandeln LargeNumber AsInteger

Optional. Gibt an, ob große Zahlen als behandelt werden sollenInteger. Sie werden BigDecimal standardmäßig wie behandelt.

Typ: Boolesch

Standard: false

unload.chunkSize

Optional. Chunk-Größe, die für das Hilfsprogramm zum Entladen verwendet wird.

Typ: Zahl

Standard: 0

## unload.ColumnFiller

Optional. Das Hilfsprogramm zum Entladen der Spalte.

Typ: Zeichenfolge

Standard: Leerzeichen

## unload.fetchSize

Optional. Ermöglicht die Einstellung der Abrufgröße bei der Bearbeitung von Cursors im Hilfsprogramm zum Entladen.

Typ: Zahl

Standard: 0

## unload.format.date

Optional. Wenn aktiviert, `unload.useDatabaseConfiguration` das Datumsformat, das im Entladeprogramm verwendet werden soll. Weitere Informationen finden Sie unter [unload.format.date](#).

Typ: Zeichenfolge

Standard: MM/TT/JJJJ

## unload.format.time

Optional. Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitformat, das im Entladeprogramm verwendet werden soll.

Typ: Zeichenfolge

Standard: HH.MM.ss

## unload.format.timestamp

Optional. Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitstempelformat, das im Entladeprogramm verwendet werden soll.

Typ: Zeichenfolge

Standard: yyyy-MM-dd-hh.mm.ss.ssssssss

### `.nbi.when entladen NotNull`

Optional. Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank nicht Null ist.

Typ: hexadezimal

Standard: 00

### `unload.nbi.WhenNull`

Optional. Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank Null ist.

Typ: hexadezimal

Standard: 6F

### `.nbi.write entladen NullIndicator`

Optional. Gibt an, ob der Null-Indikator in der Entladeausgabedatei ausgegeben werden soll.

Typ: Boolesch

Standard: false

### `unload.sql Shift CodePoint`

Optional. Das SQL-Code-Hilfsprogramm Pointshift for Unload. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank von DB2 Postgresql ist.

Typ: Zahl

Standard: 0

### `entladen.verwenden DatabaseConfiguration`

Optional. Gibt an, ob die Datums- oder Uhrzeitkonfiguration aus `application-main.yml` im Unload-Hilfsprogramm verwendet werden soll.

Typ: Boolesch

Standard: false

## unload.var Null Charls

Optional. Verwenden Sie diesen Parameter im INFTILB-Programm. Wenn er auf gesetzt ist, `true` geben alle Felder, die keine NULL-Werte zulassen, mit leeren Werten (Leerzeichen) eine leere Zeichenfolge zurück.

Typ: Boolesch

Standard: false

## Andere Eigenschaften

### qtemp.cleanup.threshold.hours

Optional. Um anzugeben, wann aktiviert ist. `qtemp.dblog` Die Lebensdauer der DB-Partition (in Stunden).

Typ: Zahl

Standard: 0

### qtemp.dblog

Optional. Ob die QTEMP-Datenbankprotokollierung aktiviert werden soll.

Typ: Boolesch

Standard: false

### qtemp.uuid.length

Optional. Die Länge der eindeutigen QTEMP-ID.

Typ: Zahl

Standard: 9

### quartz.scheduler.standby-if-error

Optional. Gibt an, ob die Auftragsausführung ausgelöst werden soll, wenn sich der Job-Scheduler im Standby-Modus befindet. Falls wahr, wird die Jobausführung nicht ausgelöst, wenn diese Option aktiviert ist.

Typ: Boolesch

Standard: false

## warm UpCache

Optional. Gibt an, ob beim Serverstart alle Daten der Datenkommunikationstabelle in einen Warmstart-Cache geladen werden sollen.

Typ: Boolesch

Standard: false

# AWS Anwendungsdefinitionsreferenz für Mainframe-Modernisierung

In AWS Mainframe Modernization konfigurieren Sie migrierte Mainframe-Anwendungen in einer JSON-Datei mit Anwendungsdefinitionen, die für die von Ihnen gewählte Runtime-Engine spezifisch ist. Eine Anwendungsdefinition enthält sowohl allgemeine als auch modulspezifische Informationen. In diesem Thema werden sowohl die Anwendungsdefinitionen von AWS Blu Age als auch von Micro Focus beschrieben und alle erforderlichen und optionalen Elemente beschrieben.

## Themen

- [Allgemeiner Kopfbereich](#)
- [Überblick über den Definitionsbereich](#)
- [AWS Beispiel für eine Blu-Age-Anwendungsdefinition](#)
- [AWS Einzelheiten zur Blu Age-Definition](#)
- [Micro Focus-Anwendungsdefinition](#)
- [Einzelheiten zur Micro Focus-Definition](#)

## Allgemeiner Kopfbereich

Jede Anwendungsdefinition beginnt mit allgemeinen Informationen über die Vorlagenversion und die Quellverzeichnisse. Die aktuelle Version der Anwendungsdefinition ist 2.0.

Verwenden Sie die folgende Struktur, um die Vorlagenversion und die Quellverzeichnisse anzugeben.

```
"template-version": "2.0",  
  "source-locations": [  
    {
```

```
    "source-id": "s3-source",
    "source-type": "s3",
    "properties": {
      "s3-bucket": "mainframe-deployment-bucket",
      "s3-key-prefix": "v1"
    }
  }
]
```

### Note

Sie können die folgende Syntax verwenden, wenn Sie S3-ARN als s3-Bucket eingeben möchten:

```
"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"
    }
  }
]
```

## Vorlagenversion

Erforderlich Gibt die Version der Anwendungsdefinitionsdatei an. Ändern Sie diesen Wert nicht. Derzeit ist der einzig zulässige Wert 2,0. Geben Sie `template-version` mit einer Zeichenfolge an.

## Quellorte

Gibt die Speicherorte der Dateien und anderer Ressourcen an, die die Anwendung zur Laufzeit benötigt.

## Quell-ID

Gibt einen Namen für den Standort an. Dieser Name wird verwendet, um bei Bedarf auf den Quellspeicherort in der JSON-Anwendungsdefinition zu verweisen.

## Quellentyp

Gibt den Typ der Quelle an. Derzeit ist der einzig zulässige Wert s3.

## Eigenschaften

Stellt die Details des Quellspeicherorts bereit. Jede Eigenschaft wird mit einer Zeichenfolge angegeben.

- `s3-bucket`- Erforderlich. Gibt den Namen des Amazon S3 S3-Buckets an, in dem die Dateien gespeichert sind.
- `s3-key-prefix`- Erforderlich. Gibt den Namen des Ordners im Amazon S3 S3-Bucket an, in dem die Dateien gespeichert sind.

## Überblick über den Definitionsbereich

Gibt die Ressourcendefinitionen der Dienste, Einstellungen, Daten und anderer typischer Ressourcen an, die die Anwendung zum Ausführen benötigt. Wenn Sie eine Anwendungsdefinition aktualisieren, erkennt AWS Mainframe Modernization Änderungen, indem es die `definition` Listen `source-locations` und aus der vorherigen und der aktuellen Version der JSON-Datei mit der Anwendungsdefinition vergleicht.

Der Definitionsabschnitt ist modulspezifisch und kann sich ändern. In den folgenden Abschnitten werden beispielhafte Engine-spezifische Anwendungsdefinitionen für beide Engines gezeigt.

## AWS Beispiel für eine Blu-Age-Anwendungsdefinition

```
{
  "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 Einzelheiten zur Blu Age-Definition

### Zuhörer — erforderlich

Geben Sie den Port an, den Sie für den Zugriff auf die Anwendung über das von AWS Mainframe Modernization erstellte Elastic Load Balancing verwenden möchten. Verwenden Sie die folgende Struktur:

```
"listeners": [{
    "port": 8194,
    "type": "http"
}],
```

### port

Erforderlich Sie können jeden verfügbaren Port mit Ausnahme der bekannten Ports 0 bis 1023 verwenden. Wir empfehlen, den Bereich von 8192 bis 8199 zu verwenden. Stellen Sie sicher, dass an diesem Port keine anderen Listener oder Anwendungen laufen.



## Typ

Erforderlich Derzeit wird nur `http` unterstützt.

## AWS Blu Age-Anwendung — erforderlich

Geben Sie mithilfe der folgenden Struktur den Speicherort an, an dem die Engine die Anwendungs-Image-Datei abruft.

```
"ba-application": {
  "app-location": "${s3-source}/murachs-v6/",
  "files-directory": "/m2/mount/myfolder",
  "enable-jics": <true|false>,
  "shared-app-location": "${s3-source}/shared/"
},
```

## Speicherort der App

Der spezifische Speicherort in Amazon S3, an dem die Anwendungs-Image-Datei gespeichert ist.

## Dateiverzeichnis

Optional. Der Speicherort der Eingabe-/Ausgabedateien für Batches. Muss ein Unterordner der Amazon EFS- oder Amazon FSx-Mount-Point-Einrichtung auf Umgebungsebene sein. Der Unterordner muss einem geeigneten Benutzer gehören, damit er von der Blu Age-Anwendung verwendet werden kann, die innerhalb AWS von Mainframe Modernization ausgeführt wird. Um dies zu erreichen, 3001 müssen beim Anhängen des Laufwerks an eine Linux-Amazon-EC2-Instance eine Gruppe mit ID 101 und ein Benutzer mit ID erstellt werden, und der gewünschte Ordner muss diesem Benutzer gehören. Auf diese Weise kann der *testclient* Ordner beispielsweise von Blu Age AWS Mainframe Modernization Managed verwendet werden.

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

## Enable-Mics

Optional. Gibt an, ob JICS aktiviert werden soll. Standardwert ist „true“. Wenn Sie diesen Wert auf False setzen, wird verhindert, dass die JICS-Datenbank erzeugt wird.

## shared-app-location

Optional. Weiterer Ort in Amazon S3, an dem gemeinsam genutzte Anwendungselemente gespeichert werden. Es kann dieselbe Art von Anwendungsstruktur wie App-Location enthalten.

## BluSAM — fakultativ

Geben Sie die BluSAM-Datenbank und den Redis-Cache mithilfe der folgenden Struktur an.

```
"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

Gibt die Eigenschaften der Datenbank an, die mit der Anwendung verwendet wird. Die Datenbank muss eine Aurora PostgreSQL-Datenbank sein. Sie können die folgenden Eigenschaften angeben:

- `nb-threads`- Fakultativ. Gibt an, wie viele dedizierte Threads für den Write-Behind-Mechanismus verwendet werden, auf den sich die BluSAM-Engine stützt. Der Standardwert ist 8.
- `batch-size`- Fakultativ. Gibt den Schwellenwert an, den der Write-Behind-Mechanismus verwendet, um Batch-Speichervorgänge zu starten. Der Schwellenwert stellt die Anzahl der geänderten Datensätze dar, die einen Batch-Speichervorgang starten, um sicherzustellen, dass geänderte Datensätze dauerhaft gespeichert werden. Der Trigger selbst basiert auf einer Kombination aus Batch-Größe und einer verstrichenen Zeit von einer Sekunde, je nachdem, welcher Wert zuerst erreicht wird. Der Standardwert ist 10 000.

- `name`- Fakultativ. Gibt den Namen der Datenbank an.
- `secret-manager-arn`— Gibt den Amazon-Ressourcennamen (ARN) des Geheimnisses an, das die Datenbankanmeldedaten enthält. Weitere Informationen finden Sie unter [Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel](#).

## Redis

Gibt die Eigenschaften des Redis-Caches an, den die Anwendung verwendet, um temporäre Daten, die sie benötigt, an einem zentralen Ort zu speichern, um die Leistung zu verbessern. Wir empfehlen, den Redis-Cache sowohl zu verschlüsseln als auch mit einem Passwort zu schützen.

- `hostname`- Gibt den Speicherort des Redis-Cache an.
- `port`- Gibt den Port an, normalerweise 6379, über den der Redis-Cache Kommunikation sendet und empfängt.
- `useSsl`— Gibt an, ob der Redis-Cache verschlüsselt ist. Wenn der Cache nicht verschlüsselt ist, setzen Sie ihn `useSsl` auf `False`.
- `secret-manager-arn`— Gibt den Amazon-Ressourcennamen (ARN) des Geheimnisses an, das das Redis-Cache-Passwort enthält. Wenn der Redis-Cache nicht kennwortgeschützt ist, geben Sie dies nicht an. `secret-manager-arn` Weitere Informationen finden Sie unter [Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel](#).

## AWS Blu Age-Nachrichtenwarteschlangen — optional

Geben Sie die JMS-MQ-Verbindungsdetails für die AWS Blu Age-Anwendung an.

```
"message-queues": [  
  {  
    "product-type": "JMS-MQ",  
    "queue-manager": "QMgr1",  
    "channel": "mqChannel1",  
    "hostname": "mqserver-host1",  
    "port": 1414,  
    "user-id": "app-user1",  
    "secret-manager-arn": "arn:aws:secretsmanager:us-  
west-2:123456789012:secret:sample/mq/test-279PTa"  
  },  
  {  
    "product-type": "JMS-MQ",  
    "queue-manager": "QMgr2",  
    "channel": "mqChannel2",
```

```
    "hostname": "mqserver-host2",
    "port": 1412,
    "user-id": "app-user2",
    "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  }
]
```

## Produkttyp

Erforderlich Gibt den Produkttyp an. Derzeit kann dies nur „JMS-MQ“ für AWS Blu Age-Anwendungen sein.

## Warteschlangenmanager

Erforderlich Gibt den Namen des Warteschlangen-Managers an.

channel

Erforderlich Gibt den Namen des Serververbindungskanals an.

hostname

Erforderlich Gibt den Hostnamen des Message Queue-Servers an.

port

Erforderlich Gibt die Listener-Port-Nummer an, auf der der Server lauscht.

## Benutzer-ID

Optional. Gibt die Benutzerkonto-ID an, die Nachrichtenwarteschlangenoperationen auf dem angegebenen Kanal ausführen darf.

secret-manager-arn

Optional. Gibt den Amazon-Ressourcennamen (ARN) von Secrets Manager an, der das Passwort des angegebenen Benutzers bereitstellt.

## AWS EFS-Konfiguration für Blu-Age-Anwendungsspeicher — optional

Geben Sie die EFS-Zugriffspunktdetails für den Anwendungsspeicher mithilfe der folgenden Struktur an.

```
"ba-application": {
  "file-permission-mask": "UMASK002"
```

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

### Dateisystem-ID

Erforderlich Die ID des EFS-Dateisystems, für das sich der Access Point bezieht. Muster: „fs-([0-9a-f] {8,40}) {1,128} \$“

### Befestigungspunkt

Erforderlich Der Einhängpunkt für das Dateisystem auf Anwendungsebene. Dieser muss sich vom Speichermountpunkt auf Umgebungsebene unterscheiden.

### Zugriffspunkt-ID

Erforderlich Die ID des Zugangspunkts, die von Amazon EFS zugewiesen wird. Muster: „^fsap-([0-9a-f] {8,40}) {1,128} \$“

### file-permission-mask

Optional. Definiert die Dateierstellungsmaske für Dateien, die durch den Anwendungsprozess erstellt wurden. Wenn der Wert beispielsweise auf gesetzt ist `UMASK006`, verfügen alle Dateien über die Berechtigung 660. Das bedeutet, dass nur der Dateibesitzer und die Dateigruppe Lese- und Schreibzugriff haben, während andere Benutzer keine Rechte haben.

#### Note

Der für dieses Feld festgelegte Wert wird nur berücksichtigt, wenn EFS-Speicher auf Anwendungsebene verwendet wird.

#### Note

Wenn die EFS-Konfiguration bereitgestellt wird, muss das Dateiverzeichnis im Abschnitt mit der Anwendungsdefinition angegeben werden. Es muss sich um einen Unterordner des Amazon EFS-Bereitstellungspunkts handeln, der auf Anwendungsebene eingerichtet wurde.

## Micro Focus-Anwendungsdefinition

Der folgende Abschnitt mit einer Beispieldefinition bezieht sich auf die Micro Focus Runtime Engine und enthält sowohl erforderliche als auch optionale Elemente.

```
{
  "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": "ADRDSU"}
    ]
  },
  "cics-settings": {
    "binary-file-location": "${s3-source}/cics/binaries",
    "csd-file-location": "${s3-source}/cics/def",
    "system-initialization-table": "BNKCICV"
  },
  "xa-resources" : [{
    "name": "XASQL",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
    "module": "${s3-source}/xa/ESPGSQLXA64.so"
  }],
  "runtime-settings": {
    "environment-variables": {
      "ES_JES_RESTART": "N",
      "EFS_MOUNT": "/m2/mount/efs"
    }
  }
}

```

## Einzelheiten zur Micro Focus-Definition

Der Inhalt des Definitionsabschnitts der Micro Focus-Anwendungsdefinitionsdatei variiert je nach den Ressourcen, die Ihre migrierte Mainframe-Anwendung zur Laufzeit benötigt.

### Listener — erforderlich

Geben Sie einen Listener mit der folgenden Struktur an:

```

"listeners": [{
  "port": 5101,
  "type": "tn3270"
}],

```

## port

Für tn3270 ist der Standardwert 5101. Bei anderen Typen von Service-Listenern variiert der Port. Sie können jeden verfügbaren Port verwenden, mit Ausnahme der bekannten Ports von 0 bis 1023. Jeder Listener sollte einen eigenen Port haben. Listener sollten sich keine Ports teilen. Weitere Informationen finden Sie unter [Listener Control](#) in der Micro Focus Enterprise Server-Dokumentation.

## Typ

Gibt den Typ des Service-Listeners an. Weitere Informationen finden Sie unter [Listeners](#) in der Micro Focus Enterprise Server-Dokumentation.

## Speicherorte von Datensätzen — erforderlich

Geben Sie den Speicherort des Datensatzes anhand der folgenden Struktur an.

```
"dataset-location": {
  "db-locations": [{
    "name": "Database1",
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456"
  }],
}
```

## DB-Standorte

Gibt den Speicherort der Datensätze an, die die migrierte Anwendung erstellt. Derzeit unterstützt AWS Mainframe Modernization nur Datensätze aus einer einzigen VSAM-Datenbank.

- **name**— Gibt den Namen der Datenbankinstanz an, die die Datensätze enthält, die die migrierte Anwendung erstellt.
- **secret-manager-arn**— Gibt den Amazon-Ressourcenamen (ARN) des Geheimnisses an, das die Datenbankanmeldedaten enthält.

## Amazon Cognito Cognito-Authentifizierungs- und Autorisierungshandler — optional

AWS Mainframe Modernization verwendet Amazon Cognito für die Authentifizierung und Autorisierung von migrierten Anwendungen. Geben Sie den Amazon Cognito Cognito-Authentifizierungshandler mithilfe der folgenden Struktur an.



```
"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"
}
```

## Benutzer-Pool-ID

Gibt den Amazon Cognito Cognito-Benutzerpool an, den AWS Mainframe Modernization verwendet, um Benutzer der migrierten Anwendung zu authentifizieren. Der Benutzerpool sollte mit dem AWS-Region für die AWS-Region Mainframe-Modernisierungsanwendung übereinstimmen. AWS

## Client-ID

Gibt die migrierte Anwendung an, auf die der authentifizierte Benutzer zugreifen kann.

## Identitätspool-ID

Gibt den Amazon Cognito Cognito-Identitätspool an, in dem der authentifizierte Benutzer ein Benutzerpool-Token gegen Anmeldeinformationen eintauscht, die dem Benutzer den Zugriff auf AWS Mainframe Modernization ermöglichen. Der AWS-Region für den Identitätspool sollte dem für die AWS-Region AWS Mainframe-Modernisierungsanwendung entsprechen.

## LDAP- und Active Directory-Handler — optional

Sie können Ihre Anwendung in Active Directory (AD) oder einen beliebigen LDAP-Server integrieren, sodass Benutzer der Anwendung ihre LDAP/AD-Anmeldeinformationen für die Autorisierung und Authentifizierung verwenden können.

Um Ihre Anwendung in AD zu integrieren

1. Folgen Sie den Schritten, die in [der Micro Focus Enterprise Server-Dokumentation unter Konfiguration von Active Directory für Enterprise Server Security](#) beschrieben sind.
2. Erstellen Sie ein AWS Secrets Manager Geheimnis mit Ihren AD/LDAP-Details für jeden AD/LDAP-Server, den Sie mit Ihrer Anwendung verwenden möchten. Informationen zum Erstellen eines Secrets finden Sie unter [Create an AWS Secrets Manager Secrets](#) Secrets im AWS Secrets Manager Benutzerhandbuch. Wählen Sie als Geheimtyp die Option Anderer Geheimtyp aus und geben Sie die folgenden Schlüssel-Wert-Paare an.

```
{
  "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>"
}
```

### Sicherheitsempfehlungen

- Denn `connectionPath` AWS Mainframe Modernization unterstützt die Protokolle LDAP und LDAP over SSL (LDAPS). Wir empfehlen die Verwendung von LDAPS, da es sicherer ist und verhindert, dass Anmeldeinformationen bei Netzwerkübertragungen auftauchen.
- Für `authorizedId` und empfehlen wir `password`, dass Sie die Anmeldeinformationen eines Benutzers angeben, der nicht mehr als die restriktivsten Lese- und Überprüfungsberechtigungen besitzt, die für die Ausführung Ihrer Anwendung erforderlich sind.
- Wir empfehlen, die AD/LDAP-Anmeldeinformationen regelmäßig zu wechseln.
- Erstellen Sie keine AD-Benutzer mit dem Benutzernamen `awsuser` oder `mfuser`. Diese beiden Benutzernamen sind für die AWS Verwendung reserviert.

Im Folgenden wird ein Beispiel gezeigt.

```
{
  "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"
}
```

Erstellen Sie das Geheimnis mit einem vom Kunden verwalteten KMS-Schlüssel. Sie müssen AWS Mainframe Modernization die DescribeSecret Berechtigungen GetSecretValue und Berechtigungen für den geheimen Schlüssel Decrypt sowie die DescribeKey Berechtigungen für den KMS-Schlüssel erteilen. Weitere Informationen finden Sie im AWS Secrets Manager Benutzerhandbuch unter [Berechtigungen für den KMS-Schlüssel](#).

3. Fügen Sie Ihrer Anwendungsdefinition Folgendes hinzu.

```
"ldap-ad-auth-handler": {
  "ldap-ad-connection-secrets": [LIST OF AD/LDAP SECRETS]
}
```

Im Folgenden wird ein Beispiel gezeigt.

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

Der LDAP/AD-Authentifizierungshandler ist für Micro Focus 8.0.11 und spätere Versionen verfügbar.

## Batch-Einstellungen — erforderlich

Geben Sie anhand der folgenden Struktur die Details an, die für die Batch-Jobs erforderlich sind, die als Teil der Anwendung ausgeführt werden.

```
"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": "ADRDSU"}
  ]
}
```

```

    ]
  }

```

## Initiatoren

Gibt einen Batch-Initiator an, der gestartet wird, wenn die migrierte Anwendung erfolgreich gestartet wird, und weiterhin ausgeführt wird, bis die Anwendung beendet wird. Sie können eine oder mehrere Klassen pro Initiator definieren. Sie können auch mehrere Initiatoren definieren. Beispielsweise:

```

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

```

Weitere Informationen finden Sie unter [So definieren Sie einen Batch-Initiator oder einen Drucker \(SEP\)](#) in der Micro Focus Enterprise Server-Dokumentation.

- `classes`- Gibt die Jobklassen an, die der Initiator ausführen kann. Sie können bis zu 36 Zeichen verwenden. Sie können die folgenden Zeichen verwenden: A-Z oder 0-9.
- `description`- Beschreibt, wofür der Initiator gedacht ist.

## Speicherort der JCL-Datei

Gibt den Speicherort der JCL-Dateien (Job Control Language) an, die für die Batch-Jobs benötigt werden, die die migrierte Anwendung ausführt.

## Programmpfad

Gibt den Pfad an, der zum Ausführen von Batch-Jobs erforderlich ist, wenn sich ein Programm in einer JCL nicht am Standardspeicherort befindet. Die verschiedenen Pfadnamen werden durch einen Doppelpunkt (:) getrennt.

**Note**

Der Programmpfad kann nur ein EFS-Pfad sein.

## Systemprozedur-Bibliotheken

Gibt die partitionierten Standarddatensätze an, die nach JCL-Prozeduren durchsucht werden. Die Prozedur ist jedoch nicht in der JCL oder über die JCLLIB-Anweisungen zu finden. Diese Datensätze müssen katalogisiert werden und der Katalogname muss verwendet werden. Und die Einträge werden durch ein Semikolon (;) getrennt.

## Aliase

Definiert eine Zuordnung der in JCL verwendeten Utility- und Programmnamen zum Implementierungsnamen des Utilities. AWS und Batch-Dienstprogramme von Drittanbietern (z. B. M2SFTP, M2WAIT, Syncsort usw.) können optional Aliase haben, sodass die JCL nicht geändert werden muss. Beispielsweise:

- FDSSORT Alias FDSSORT für SORT und Alias FDSICET für ICETOOL
- ADRDSSU Alias MFADRDSU für ADRDSSU
- Synchronisieren Sie den Sortenalias DMXMFSTR für SORT

## CICS-Einstellungen — erforderlich

Geben Sie mithilfe der folgenden Struktur die Details an, die für die CICS-Transaktionen erforderlich sind, die als Teil der Anwendung ausgeführt werden.

```
"cics-settings": {
  "binary-file-location": "${s3-source}/cics/binaries",
  "csd-file-location": "${s3-source}/cics/def",
  "system-initialization-table": "BNKCICV"
}
```

## Speicherort der Binärdatei

Gibt den Speicherort der CICS-Transaktionsprogrammdateien an.

## Speicherort der CSD-Datei

Gibt den Speicherort der CICS-Ressourcendefinitionsdatei (CSD) für diese Anwendung an. Weitere Informationen finden Sie unter [CICS-Ressourcendefinitionen](#) in der Micro Focus Enterprise Server-Dokumentation.

## Systeminitialisierungstabelle

Gibt die Systeminitialisierungstabelle (SIT) an, die die migrierte Anwendung verwendet. Der Name der SIT-Tabelle kann bis zu 8 Zeichen lang sein. Sie können A-Z, 0-9, \$, @ und # verwenden. Weitere Informationen finden Sie unter [CICS-Ressourcendefinitionen](#) in der Micro Focus Enterprise Server-Dokumentation.

## XA-Ressourcen — erforderlich

Geben Sie die erforderlichen Details für die XA-Ressourcen, die die Anwendung benötigt, mithilfe der folgenden Struktur an.

```
"xa-resources" : [{  
    "name": "XASQL",  
    "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",  
    "module": "${s3-source}/xa/ESPGSQLXA64.so"  
}]
```

### Name

Erforderlich Gibt den Namen der XA-Ressource an.

### secret-manager-arn

Gibt den Amazon-Ressourcennamen (ARN) für das Geheimnis an, das die Anmeldeinformationen für die Verbindung mit der Datenbank enthält.

### Modul

Gibt den Speicherort der ausführbaren Datei des RM-Switch-Moduls an. Weitere Informationen finden Sie unter [Planung und Entwurf von XARs](#) in der Micro Focus Enterprise Server-Dokumentation.

## Laufzeiteinstellungen — optional

Geben Sie die Details an, die für die Laufzeiteinstellungen zur Verwaltung zulässiger Umgebungsvariablen erforderlich sind, indem Sie die folgende Struktur verwenden.

```
"runtime-settings": {
  "environment-variables": {
    "ES_JES_RESTART": "N",
    "EFS_MOUNT": "/m2/mount/efs"
  }
}
```

### Umgebungsvariablen

Gibt von Micro Focus unterstützte Umgebungsvariablen an, die auf die Laufzeit dieser Anwendung angewendet werden.

- `ES_JES_RESTART` ist eine Micro Focus-Umgebungsvariable, die die JCL-Neustartverarbeitung ermöglicht. Optional können Sie sie auch `ES_ALLOC_OVERRIDE` als Micro Focus-Umgebungsvariable verwenden.
- `EFS_MOUNT` ist eine benutzerdefinierte Umgebungsvariable, die Ihre Anwendung möglicherweise verwendet, um zu identifizieren, wo sich der EFS-Mount der Umgebung befindet.

Sie können auf alle [Micro Focus-Umgebungsvariablen](#) im Handbuch Micro Focus Enterprise Server für UNIX zugreifen.

## AWS Referenz zur Definition von Datensätzen zur Mainframe-Modernisierung

Wenn Ihre Anwendung mehr als ein paar Datensätze für die Verarbeitung benötigt, ist es ineffizient, sie nacheinander in die AWS Mainframe-Modernisierungskonsole einzugeben. Stattdessen empfehlen wir, dass Sie eine JSON-Datei erstellen, um jeden Datensatz zu spezifizieren. Verschiedene Datensatztypen werden im JSON unterschiedlich angegeben, obwohl viele Parameter gemeinsam sind. In diesem Dokument werden die Einzelheiten der JSON-Datei beschrieben, die für den Import verschiedener Arten von Datensätzen erforderlich ist.

**Note**

Bevor Sie Datensätze importieren, müssen Sie die Datensätze vom Mainframe auf AWS übertragen. Dann müssen Sie sicherstellen, dass die Datensätze vom Mainframe-Format in ein Format konvertiert werden, das verwendet werden kann. Transformieren Sie die Daten bei Bedarf nach Bedarf und speichern Sie die transformierten Datensätze in Amazon S3. Geben Sie den Namen des Buckets und des Ordners in der JSON-Datei mit der Datensatzdefinition an.

Wenn Sie die Micro Focus Runtime Engine verwenden, können Sie das DFCONV Hilfsprogramm verwenden, um die Datensätze zu konvertieren. Wir nehmen dieses Tool in unsere Micro Focus Enterprise Developer- und Enterprise Server-Images auf. Weitere Informationen finden Sie unter [DFCONV Batch File Conversion](#) in der Micro Focus Enterprise Developer-Dokumentation.

**Themen**

- [Gemeinsame Eigenschaften](#)
- [Beispiel für ein Datensatz-Anforderungsformat für VSAM](#)
- [Beispiel für ein Datensatz-Anforderungsformat für GDG Base](#)
- [Beispiel für ein Anforderungsformat für Datensätze für PS- oder GDG-Generationen](#)
- [Beispiel für ein Datensatz-Anforderungsformat für PO](#)

**Gemeinsame Eigenschaften**

Allen Datensätzen sind mehrere Parameter gemeinsam. Diese Parameter decken die folgenden Bereiche ab:

- Informationen über den Datensatz (`datasetName`, `datasetOrg`, `recordLength`, `encoding`)
- Informationen über den Speicherort, von dem Sie importieren, d. h. den Quellspeicherort des Datensatzes. Dies ist nicht der Standort auf dem Mainframe. Dies ist der Pfad zum Amazon S3 S3-Speicherort, an den Sie den Datensatz hochgeladen haben (`externalLocation`).
- Informationen über den Speicherort, an den Sie importieren, d. h. den Zielort des Datensatzes. Dieser Speicherort ist entweder eine Datenbank oder ein Dateisystem, abhängig von Ihrer Runtime-Engine. (`storageType` und `relativePath`).
- Informationen zum Datensatztyp (spezifischer Datensatztyp, Format, Kodierung usw.).



Jede Datensatzdefinition hat dieselbe JSON-Struktur. Das folgende JSON-Beispiel zeigt all diese gemeinsamen Parameter.

```
{
  "dataSet": {
    "storageType": "Database",
    "datasetName": "MFI01V.MFIDEMO.BNKACC",
    "relativePath": "DATA",
    "datasetOrg": {
      "type": {
        type-specific properties
        ...
      },
    },
  },
},
}
```

Die folgenden Eigenschaften sind allen Datensätzen gemeinsam.

### storageType

Erforderlich Gilt für den Zielstandort. Gibt an, ob der Datensatz in einer Datenbank oder einem Dateisystem gespeichert ist. Die möglichen Wert sind Database oder FileSystem.

- AWS Blu Age Runtime Engine: Dateisysteme werden nicht unterstützt. Sie müssen eine Datenbank verwenden.
- Micro Focus Runtime Engine: Sowohl Datenbanken als auch Dateisysteme werden unterstützt. Sie können entweder Amazon Relational Database Service oder Amazon Aurora für Datenbanken und Amazon Elastic File System oder Amazon FSx for Lustre für Dateisysteme verwenden.

### datasetName

Erforderlich Gibt den vollqualifizierten Namen des Datensatzes an, so wie er auf dem Mainframe angezeigt wird.

### RelativePath

Erforderlich Gilt für den Zielort. Gibt den relativen Speicherort des Datensatzes in der Datenbank oder im Dateisystem an.

## DatasetOrg

Erforderlich Gibt den Typ des Datensatzes an. Mögliche Werte sind vsam, gdg, ps, po, oder unknown.

- AWS Blu Age Runtime Engine: Es werden nur Datensätze vom Typ VSAM unterstützt.
- Micro Focus Runtime Engine: Datensätze vom Typ VSAM, GDG, PS, PO oder Unknown werden unterstützt.

### Note

Wenn Ihre Anwendung Dateien benötigt, bei denen es sich nicht um COBOL-Datendateien, sondern um PDF- oder andere Binärdateien handelt, können Sie diese wie folgt angeben:

```
"datasetOrg": {
  "type": PS {
    "format": U
  },
}
```

## Beispiel für ein Datensatz-Anforderungsformat für VSAM

- AWS Blu Age Runtime Engine: unterstützt.
- Micro Focus Runtime Engine: unterstützt.

Wenn Sie VSAM-Datensätze importieren, geben Sie vsam als andatasetOrg. Ihr JSON sollte dem folgenden Beispiel ähneln:

```
{
  "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"
}

```

Die folgenden Eigenschaften werden für VSAM-Datensätze unterstützt.

### encoding

Erforderlich Gibt die Zeichensatzkodierung des Datensatzes an. Mögliche Werte sind ASCII (A), EBCDIC (E) und Unknown (). ?

### Format

Erforderlich Gibt den VSAM-Datensatztyp und das Datensatzformat an.

- AWS Blu Age Runtime Engine: Mögliche Werte sind ESDS (ES), KSDS (KS) und RRDS (). RR Das Datensatzformat kann fest oder variabel sein.
- Micro Focus Runtime Engine: Mögliche Werte sind ESDS (ES), KSDS (KS) und RRDS (). RR Die VSAM-Definition beinhaltet das Datensatzformat, sodass Sie es nicht separat angeben müssen.

### Primärschlüssel

Gilt nur für VSAM KSDS-Datensätze. Gibt den Primärschlüssel an. Besteht aus dem Namen des Primärschlüssels, dem Schlüsselversatz und der Schlüssellänge. Die name sind optional; offset und length sind erforderlich.

### Länge des Datensatzes

Erforderlich Gibt die Länge eines Datensatzes an. Bei Datensatzformaten mit fester Länge müssen diese Werte übereinstimmen.

- AWS Blu Age Runtime Engine: für VSAM ESDS, KSDS und RRDS ist optional und erforderlich.  
min max

- Micro Focus Runtime Engine: `min` und `max` sind erforderlich.

## Externer Standort

Erforderlich Gibt den Quellspeicherort an, d. h. den Amazon S3 S3-Bucket, in den Sie den Datensatz hochgeladen haben.

## Engine-spezifische Eigenschaften von Blue Age

Die AWS Blu Age Runtime Engine unterstützt die Komprimierung von VSAM-Datensätzen. Das folgende Beispiel zeigt, wie Sie diese Eigenschaft in JSON angeben können.

```
{
  common properties
  ...
  "datasetOrg": {
    "vsam": {
      common properties
      ...
      "compressed": boolean,
      common properties
      ...
    }
  }
}
```

Geben Sie die Komprimierungseigenschaft wie folgt an:

## Kompression

Optional. Gibt an, ob Indizes für diesen Datensatz als komprimierte Werte gespeichert werden. Wenn Sie über einen großen Datensatz verfügen (normalerweise > 100 MB), sollten Sie dieses Kennzeichen auf `true` setzen.

## Beispiel für ein Datensatz-Anforderungsformat für GDG Base

- AWS Blu Age Runtime Engine: wird nicht unterstützt.
- Micro Focus Runtime Engine: unterstützt.

Wenn Sie GDG-Basisdatensätze importieren, geben Sie `gdg` als `andatasetOrg`. Ihr JSON-Code sollte dem folgenden Beispiel ähneln:

```
{
  "storageType": "Database",
  "datasetName": "AWS.M2.GDG",
  "relativePath": "DATA",
  "datasetOrg": {
    "gdg": {
      "limit": "3",
      "rollDisposition": "Scratch and No Empty"
    }
  }
}
```

Die folgenden Eigenschaften werden für GDG-Basisdatensätze unterstützt.

#### limit

Erforderlich Gibt die Anzahl der aktiven Generationen oder Verzerrungen an. Für einen GDG-Basiscluster ist der Höchstwert 255.

#### RollDisposition

Optional. Gibt an, wie Generierungsdatensätze behandelt werden sollen, wenn das Maximum erreicht oder überschritten wird. Mögliche Werte sind `No Scratch and No Empty`, `Scratch and No Empty`, `Scratch and Empty` oder `No Scratch and Empty`. Der Standardwert ist `Scratch and No Empty`.

## Beispiel für ein Anforderungsformat für Datensätze für PS- oder GDG-Generationen

- AWS Blu Age Runtime Engine: wird nicht unterstützt.
- Micro Focus Runtime Engine: unterstützt.

Wenn Sie Datensätze der PS- oder GDG-Generation importieren, geben Sie `ps` als `andatasetOrg`. Ihr JSON-Code sollte dem folgenden Beispiel ähneln:

```
{
```

```

"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"
}
}

```

Die folgenden Eigenschaften werden für Datensätze der PS- oder GDG-Generation unterstützt.

#### Format

Erforderlich Gibt das Format der Datensatzdatensätze an. Mögliche Werte sind F, FAFB, FBA, FBM, FBS, FM, FS, LSEQ, U, V, VA, VBVBA, VBM, VBS, VM, und VS.

#### encoding

Erforderlich Gibt die Zeichensatzkodierung des Datensatzes an. Mögliche Werte sind ASCII (A), EBCDIC (E) und Unknown () ?

#### Länge des Datensatzes

Erforderlich Gibt die Länge eines Datensatzes an. Sie müssen sowohl die Mindest- (min) als auch die Höchstlänge (max) des Datensatzes angeben. Bei Datensatzformaten mit fester Länge müssen diese Werte übereinstimmen.

#### Externer Standort

Erforderlich Gibt den Quellspeicherort an, d. h. den Amazon S3 S3-Bucket, in den Sie den Datensatz hochgeladen haben.

## Beispiel für ein Datensatz-Anforderungsformat für PO

Wenn Sie PO-Datensätze importieren, geben Sie `po` als `andataset0rg`. Ihr JSON-Code sollte dem folgenden Beispiel ähneln:

```
{
  "storageType": "Database",
  "datasetName": "AWS.M2.PO.PROC",
  "relativePath": "DATA",
  "dataset0rg": {
    "po": {
      "format": "LSEQ",
      "encoding": "A",
      "memberFileExtensions": ["PRC"]
    }
  },
  "recordLength": {
    "min": 80,
    "max": 80
  }
},
"externalLocation": {
  "s3Location": "s3://$M2_DATA_STORE/source/proc/"
}
}
```

Die folgenden Eigenschaften werden für PO-Datensätze unterstützt.

### Format

Erforderlich Gibt das Format der Datensatzdatensätze an. Mögliche Werte sind `F`, `FAFB`, `FBA`, `FBM`, `FBS`, `FM`, `FS`, `LSEQ`, `U`, `V`, `VA`, `VBVBA`, `VBM`, `VBS`, `VM`, und `VS`.

### encoding

Erforderlich Gibt die Zeichensatzkodierung des Datensatzes an. Mögliche Werte sind `ASCII (A)`, `EBCDIC (E)` und `Unknown (.)`. ?

### memberFileExtensions

Erforderlich Gibt ein Array an, das eine oder mehrere Dateinamenerweiterungen enthält, sodass Sie angeben können, welche Dateien als PDS-Mitglied aufgenommen werden sollen.

## Datensatzlänge

Optional. Gibt die Länge eines Datensatzes an. Sowohl die Mindest- (min) als auch die Höchstlänge (max) des Datensatzes sind optional. Bei Datensatzformaten mit fester Länge müssen diese Werte übereinstimmen.

## Externer Standort

Erforderlich Gibt den Quellspeicherort an, d. h. den Amazon S3 S3-Bucket, in den Sie den Datensatz hochgeladen haben.

### Note

Die aktuelle Implementierung für die Micro Focus Runtime Engine fügt PDS-Einträge als dynamische Datensätze hinzu.



# Verwaltete Laufzeitumgebungen bei der AWS Mainframe-Modernisierung

Wenn Sie mit der AWS Mainframe-Modernisierung noch nicht vertraut sind, finden Sie zunächst die folgenden Themen:

- [Was ist AWS Mainframe-Modernisierung?](#)
- [Einrichtung der AWS Mainframe-Modernisierung](#)
- [Erste Schritte mit der AWS Mainframe-Modernisierung](#)
- [Tutorial: Verwaltete Laufzeit für AWS Blu Age](#)
- [Tutorial: Verwaltete Laufzeit für Micro Focus](#)

Eine Laufzeitumgebung in AWS Mainframe Modernization ist eine benannte Kombination aus AWS Rechenressourcen, einer Runtime-Engine und den von Ihnen angegebenen Konfigurationsdetails. Die Laufzeitumgebung hostet eine oder mehrere Anwendungen. Anwendungen im Rahmen der AWS Mainframe-Modernisierung enthalten migrierte Mainframe-Workloads. Sie können die Runtime-Engine für die Umgebungen wählen, die Sie erstellen. Wählen Sie AWS Blu Age, wenn Sie das automatisierte Refactoring-Muster verwenden, und Micro Focus, wenn Sie das Replatforming-Muster verwenden. Sie können auch die Menge an Rechenressourcen wählen, die für Ihre Anwendung geeignet sind, und optional Speicher an Laufzeitumgebungen anhängen. AWS Mainframe-Modernisierung ermöglicht Amazon CloudWatch Metriken und Logging für Sie, sodass Sie Ihre Laufzeitumgebung überwachen können.

## Themen

- [Erstellen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#)
- [Aktualisieren Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#)
- [Beenden Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#)
- [Starten Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung neu](#)
- [Löschen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#)

# Erstellen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine Umgebung für die AWS Mainframe-Modernisierung zu erstellen.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die unter aufgeführten Schritte abgeschlossen haben. [Einrichtung der AWS Mainframe-Modernisierung](#)

## Erstellen Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung zu erstellen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Umgebung erstellen möchten.
3. Wählen Sie auf der Seite Umgebungen die Option Umgebung erstellen aus.
4. Geben Sie auf der Seite Basisinformationen angeben die folgenden Informationen ein:
  - a. Geben Sie im Abschnitt Name und Beschreibung einen Namen für die Umgebung ein.
  - b. (Optional) Geben Sie im Feld Umgebungsbeschreibung eine Beschreibung für die Umgebung ein. Diese Beschreibung kann Ihnen und anderen Benutzern helfen, den Zweck der Laufzeitumgebung zu identifizieren.
  - c. Wählen Sie im Bereich Engine-Optionen Blu Age für automatisiertes Refactoring oder Micro Focus für Replatforming.
  - d. Wählen Sie eine Version für die Engine, die Sie ausgewählt haben.
  - e. (Optional) Wählen Sie im Abschnitt „Tags“ die Option Neues Tag hinzufügen aus, um Ihrer Umgebung ein oder mehrere Umgebungs-Tags hinzuzufügen. Ein Umgebungs-Tag ist eine benutzerdefinierte Attributbezeichnung, mit der Sie Ihre AWS Ressourcen organisieren und verwalten können.
  - f. Wählen Sie Weiter aus.
5. Geben Sie auf der Seite „Konfigurationen angeben“ die folgenden Informationen ein:
  - a. Wählen Sie im Abschnitt Verfügbarkeit die Option Standalone-Laufzeitumgebung oder Hochverfügbarkeitscluster aus.


Das Verfügbarkeitsmuster bestimmt, wie verfügbar Ihre Anwendung sein wird, wenn sie ausgeführt wird. Standalone ist für Entwicklungszwecke in Ordnung. Hochverfügbarkeit gilt für Anwendungen, die jederzeit verfügbar sein müssen.

- b. Wählen Sie unter Ressourcen einen Instanztyp und die gewünschte Kapazität aus.

Bei diesen Ressourcen handelt es sich um die von der AWS Mainframe-Modernisierung verwalteten Amazon EC2 EC2-Instances, die Ihre Laufzeitumgebung hosten werden. Standalone-Laufzeitumgebungen bieten zwei Optionen für den Instance-Typ und lassen nur eine Instanz zu. Laufzeitumgebungen mit hoher Verfügbarkeit bieten zwei Optionen für den Instanztyp und erlauben bis zu zwei Instanzen.

Weitere Informationen finden Sie unter [Amazon EC2 EC2-Instance-Typen](#). Wenn Sie Hilfe benötigen, wenden Sie sich an einen AWS Mainframe-Spezialisten.


6. Gehen Sie im Abschnitt Sicherheit und Netzwerk wie folgt vor:
  - a. Wenn Sie möchten, dass auf die Anwendungen öffentlich zugegriffen werden kann, wählen Sie In dieser Umgebung bereitgestellte Anwendungen öffentlich zugänglich machen aus.
  - b. Wählen Sie eine Virtual Private Cloud (VPC).
  - c. Wenn Sie das Hochverfügbarkeitsmuster verwenden, wählen Sie zwei oder mehr Subnetze aus. Wenn Sie das eigenständige Muster mit der AWS Blu Age-Engine verwenden, wählen Sie zwei oder mehr Subnetze aus. Wenn Sie das eigenständige Muster mit der Micro Focus Engine verwenden, können Sie ein Subnetz angeben.
  - d. Wählen Sie eine Sicherheitsgruppe für die VPC, die Sie ausgewählt haben.

 Note

AWS Mainframe Modernization erstellt einen Network Load Balancer, mit dem Sie Verbindungen zu Ihrer Laufzeitumgebung verteilen können. Stellen Sie sicher, dass Ihre Sicherheitsgruppenregeln für eingehende Nachrichten den Zugriff von einer IP-Adresse auf den Port zulassen, den Sie in der `listener` Eigenschaft der Anwendungsdefinition angegeben haben. Weitere Informationen finden Sie unter [Registrieren von Zielen](#) im Benutzerhandbuch für Network Load Balancers.

- e. Wählen Sie im Feld KMS-Schlüssel die Option Verschlüsselungseinstellungen anpassen aus, wenn Sie eine vom Kunden verwaltete AWS KMS key Verschlüsselung verwenden

möchten. Weitere Informationen finden Sie unter [Datenverschlüsselung im Ruhezustand für den AWS Mainframe-Modernisierungsservice](#).

 Note

Standardmäßig verschlüsselt AWS Mainframe Modernization Ihre Daten mit einer Datei AWS KMS key , die AWS Mainframe Modernization für Sie besitzt und verwaltet. Sie können sich jedoch dafür entscheiden, eine vom Kunden verwaltete Lösung zu verwenden. AWS KMS key

- f. (Optional) Wählen Sie einen AWS KMS key Namen oder einen Amazon-Ressourcennamen (ARN). Wählen Sie alternativ Create an, AWS KMS key um zur AWS KMS Konsole zu wechseln und eine neue AWS KMS key zu erstellen.
- g. Wählen Sie Weiter aus.
7. (Optional) Wählen Sie auf der Seite Speicher anhängen ein oder mehrere Amazon EFS- oder Amazon FSx-Dateisysteme aus und klicken Sie dann auf Weiter.
8. Wählen Sie im Abschnitt Wartungsfenster aus, wann Sie ausstehende Änderungen an der Umgebung anwenden möchten.
  - Wenn Sie „Keine Präferenz“ wählen, wählt AWS Mainframe Modernization ein optimiertes Wartungsfenster für Sie aus.
  - Wenn Sie ein bestimmtes Wartungsfenster angeben möchten, wählen Sie Neues Wartungsfenster auswählen. Wählen Sie dann einen Wochentag, eine Startzeit und eine Dauer für das Wartungsfenster.

Weitere Informationen zum Wartungsfenster finden Sie unter [AWS Wartungsfenster für die Mainframe-Modernisierung](#).

Wählen Sie Weiter aus.

9. Überprüfen Sie auf der Seite Überprüfen und erstellen die von Ihnen eingegebenen Informationen, und wählen Sie dann Umgebung erstellen aus.

# Aktualisieren Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung zu aktualisieren. Sie können die Nebenversion der Runtime-Engine oder den Instanztyp aktualisieren, der die Laufzeitumgebung hostet. Sie können wählen, ob Sie Updates sofort oder während des bevorzugten Wartungsfensters anwenden möchten.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Aktualisieren Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung zu aktualisieren

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Umgebung erstellt wurde, die Sie aktualisieren möchten.
3. Wählen Sie auf der Seite Umgebungen die Umgebung aus, die Sie aktualisieren möchten.
4. Wählen Sie auf der Detailseite für die Umgebung Aktionen und dann Umgebung bearbeiten aus.
5. Nehmen Sie eine oder alle der folgenden Änderungen vor:
  - Wählen Sie im Abschnitt Engine-Optionen die gewünschte Engine-Version aus.
  - Wählen Sie im Abschnitt Ressourcen den gewünschten Instanztyp aus.
  - Wählen Sie im Abschnitt Wartungsfenster den gewünschten Tag, die Uhrzeit und die Dauer aus.

### Note

Die einzigen Änderungen, die Sie während des Wartungsfensters übernehmen können, sind Änderungen an der Engine-Version. Sie müssen alle anderen Änderungen sofort anwenden.

6. Wählen Sie Weiter aus.

7. Wählen Sie unter Wann diese Änderungen übernommen werden sollen die Option Sofort oder Während des nächsten Wartungsfensters aus. Wählen Sie dann Umgebung aktualisieren aus.

Wenn Sie Sofort wählen, wird eine Meldung angezeigt, wenn die Aktualisierung der Umgebung abgeschlossen ist.

## AWS Wartungsfenster für die Mainframe-Modernisierung

Jede Laufzeitumgebung hat ein wöchentliches zweistündiges Wartungsfenster. Während dieser Zeit werden alle Systemänderungen übernommen. Das Wartungsfenster bietet Ihnen die Möglichkeit, zu kontrollieren, wann Änderungen vorgenommen und Software- und Sicherheitspatches installiert werden. Wenn ein Wartungsereignis für eine bestimmte Woche geplant ist, beginnt es innerhalb dieses zweistündigen Wartungsfensters. Die meisten Wartungsereignisse werden auch während des zweistündigen Wartungsfensters abgeschlossen, obwohl größere Wartungsereignisse mehr als ein paar Stunden in Anspruch nehmen können.

Das zweistündige Wartungsfenster wird nach dem Zufallsprinzip aus einem Zeitblock von 8 Stunden pro Region ausgewählt. Wenn Sie bei der Erstellung einer Laufzeitumgebung kein Wartungsfenster angeben, weist AWS Mainframe Modernization an einem zufällig ausgewählten Wochentag ein zweistündiges Wartungsfenster zu.

AWS Die Mainframe-Modernisierung verbraucht während der Wartung einige Ressourcen in Ihrer Umgebungsinstanz. Möglicherweise stellen Sie während der Wartung minimale Auswirkungen auf die Leistung oder einige Unterbrechungen der Anwendungen fest.

Die folgende Tabelle zeigt die Standardzeitblöcke, in denen Wartungsfenster für jede Region zugewiesen werden.

Name der Region	Region	Zeitblock
USA Ost (Nord-Virginia)	us-east-1	03:00 bis 11:00 Uhr UTC
USA West (Oregon)	us-west-2	06:00 - 14:00 UTC
Asia Pacific (Mumbai)	ap-south-1	06:00 - 14:00 UTC
Asien-Pazifik (Singapur)	ap-southeast-1	14:00 - 22:00 UTC

Name der Region	Region	Zeitblock
Asien-Pazifik (Sydney)	ap-southeast-2	12:00 - 20:00 UTC
Asien-Pazifik (Tokio)	ap-northeast-1	13:00 - 21:00 UTC
Canada (Central)	ca-central-1	03:00 bis 11:00 Uhr UTC
Europa (Frankfurt)	eu-central-1	21:00 - 05:00 UTC
Europa (Irland)	eu-west-1	22:00 - 06:00 UTC
Europe (London)	eu-west-2	22:00 - 06:00 UTC
Europe (Paris)	eu-west-3	23:59 - 07:29 UTC
Südamerika (São Paulo)	sa-east-1	00:00 - 08:00 UTC

## Beenden Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung zu beenden. Wenn Sie eine Umgebung beenden, werden die aktuellen Anwendungsbereitstellungen beibehalten, und Ihnen wird die Umgebung erst in Rechnung gestellt, wenn die Umgebung neu gestartet wird.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Beenden Sie eine Laufzeitumgebung

Wenn Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung beenden müssen, gehen Sie ähnlich vor wie im Abschnitt [Umgebung aktualisieren](#) beschrieben.

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung zu beenden. Wenn Sie eine Umgebung beenden, werden die

aktuellen Anwendungsbereitstellungen beibehalten, und Ihnen wird die Umgebung erst in Rechnung gestellt, wenn die Umgebung neu gestartet wird.

## Beenden Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung für die AWS Mainframe-Modernisierung zu beenden, folgen Sie ähnlichen Schritten wie im Abschnitt Umgebung aktualisieren.

### Note

Sie müssen alle Anwendungen beenden, bevor Sie die Umgebung beenden.

Um eine Laufzeitumgebung zu beenden

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Umgebung, die Sie beenden möchten, erstellt wurde.
3. Wählen Sie auf der Seite Umgebungen die Umgebung aus, die Sie beenden möchten.
4. Wählen Sie auf der Detailseite für die Umgebung Aktionen und dann Umgebung bearbeiten aus.
5. Suchen Sie auf der Seite Umgebung bearbeiten den Abschnitt Ressourcen und aktualisieren Sie die gewünschte Kapazität auf Null.

### Note

Um eine Umgebung zu beenden, können Sie nur wählen, ob Sie sofort beenden möchten.

6. Wählen Sie Weiter aus.
7. Wählen Sie unter Wann diese Änderungen übernommen werden sollen die Option Sofort aus. Wählen Sie dann Umgebung aktualisieren.

Sie erhalten eine Meldung, wenn die Umgebungskapazität aktualisiert wird.



# Starten Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung neu

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung neu zu starten. Wenn Sie eine Laufzeitumgebung neu starten, wird die Abrechnung für die Umgebung wieder aufgenommen.

## Starten Sie eine Laufzeitumgebung neu

Um eine Laufzeitumgebung für die AWS Mainframe-Modernisierung neu zu starten, folgen Sie ähnlichen Schritten wie im Abschnitt Stopp-Umgebung.

Um eine Laufzeitumgebung neu zu starten

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Umgebung erstellt wurde, die Sie neu starten möchten.
3. Wählen Sie auf der Seite Umgebungen die Umgebung aus, die Sie neu starten möchten.
4. Wählen Sie auf der Detailseite für die Umgebung Aktionen und dann Umgebung bearbeiten aus.

### Note

Die gewünschte Kapazität für eine eigenständige Umgebung kann nur auf 1 aktualisiert werden. Um eine Laufzeitumgebung neu zu starten, können Sie nur den sofortigen Neustart wählen.

5. Suchen Sie auf der Seite Umgebung bearbeiten den Abschnitt Ressourcen und aktualisieren Sie die gewünschte Kapazität von Null auf die erforderliche Kapazität.
6. Wählen Sie Weiter aus.
7. Wählen Sie unter Wann sollen diese Änderungen übernommen werden? die Option Sofort aus. Wählen Sie dann Umgebung aktualisieren.

Sie erhalten eine Meldung, wenn die Umgebungskapazität aktualisiert und die Umgebung neu gestartet wird.

# Löschen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung

Verwenden Sie die AWS Mainframe-Modernisierungskonsolle, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung zu löschen.

Diese Anleitungen setzen die Durchführung der Schritte unter [Einrichtung der AWS Mainframe-Modernisierung](#) voraus.

## Löschen Sie eine Laufzeitumgebung

Wenn Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung löschen müssen, stellen Sie sicher, dass Sie zuerst alle bereitgestellten Anwendungen aus der Umgebung löschen. Sie können keine Laufzeitumgebung löschen, in der Anwendungen bereitgestellt werden.

So löschen Sie eine Umgebung

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der die Umgebung erstellt wurde, die Sie löschen möchten.
3. Wählen Sie auf der Seite Umgebungen die Umgebung aus, die Sie löschen möchten, und klicken Sie dann auf Aktionen und Umgebung löschen.
4. Geben Sie im Fenster Umgebung löschen ein, delete um zu bestätigen, dass Sie die Laufzeitumgebung löschen möchten, und wählen Sie dann Löschen.

# Anwendungstests bei der AWS Mainframe-Modernisierung

AWS Application Testing befindet sich in der Vorschauversion für AWS Mainframe Modernization und kann sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

AWS Mainframe Modernization Application Testing bietet automatisierte funktionale Äquivalenztests für Ihre Migrationsprojekte.

## Themen

- [Was sind Anwendungstests für die AWS Mainframe-Modernisierung?](#)
- [AWS Konzepte für das Testen von Anwendungen zur Mainframe-Modernisierung](#)
- [Tutorial: Richten Sie die CardDemo Beispielanwendung ein](#)
- [Tutorial: AWS Mainframe-Modernisierung, Anwendungstests, Wiederholung und Vergleich unter Verwendung von CardDemo AWS Blu Age, bereitgestellt auf Amazon EC2](#)
- [AWS Mainframe-Modernisierung, Anwendungstests, unterstützte Datensätze, Codepages](#)

## Was sind Anwendungstests für die AWS Mainframe-Modernisierung?

AWS Application Testing befindet sich in der Vorabversion für AWS Mainframe Modernization und kann sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

Tests wirken sich erheblich auf Migrationsprojekte aus. Es kann bis zu 70% der Zeit und des Aufwands Ihres Migrations-, Modernisierungs- oder Erweiterungsprojekts in Anspruch nehmen. AWS Application Testing, eine Funktion von AWS Mainframe Modernization, bietet automatisierte funktionale Äquivalenztests für Ihre migrierten Anwendungen. Mit funktionalen Äquivalenztests können Sie überprüfen, ob Ihre Anwendungen auf dem Ihren Mainframe Ihren Anwendungen auf dem Mainframe AWS Cloud entsprechen. AWS Application Testing vergleicht automatisch Änderungen an Datensätzen, Datenbankeinträgen und Online-3270-Bildschirmen zwischen Ihrem Mainframe und. AWS Darüber hinaus ermöglichen Anwendungstests wiederholbare Tests, sodass

Sie Ihre Testszzenarien viele Male ausführen können, während Sie die Zielarchitektur aktualisieren, Probleme lösen und zu einer vollständig migrierten Anwendung übergehen. Nach der Migration können Sie Application Testing weiterhin für Regressionstests verwenden, um sicherzustellen, dass Updates für Runtime-Engines oder andere Komponenten keine Regressionen verursachen. Anwendungstests sind kosteneffizient: Zieltestumgebungen werden mithilfe der vom Benutzer bereitgestellten CloudFormation Vorlagen erstellt, wobei Infrastructure-as-Code (IaC) -Konzepte genutzt werden. Anwendungstests beschleunigen Migrationsprojekte mithilfe der Elastizität der Cloud. Sie können unabhängige Testszzenarien in beliebig vielen parallel Umgebungen ausführen und so die Testzeiten reduzieren.

## Themen

- [Verwenden Sie Application Testing zum ersten Mal?](#)
- [Vorteile von Application Testing](#)
- [Integration mit AWS CloudFormation](#)
- [Wie funktioniert das Testen von Anwendungen](#)
- [Zugehörige Services](#)
- [Zugriff auf Anwendungstests](#)
- [Preise für Anwendungstests](#)

## Verwenden Sie Application Testing zum ersten Mal?

Wenn Sie Application Testing zum ersten Mal verwenden, empfehlen wir Ihnen, zunächst die folgenden Abschnitte zu lesen:

- [Konzepte für das Testen von Anwendungen](#)
- [Tutorial: Richten Sie ein CardDemo](#)

## Vorteile von Application Testing

Anwendungstests bieten mehrere Vorteile, die Sie bei Ihrem Migrationsprozess unterstützen:

- Beschleunigung, Agilität und Flexibilität beim Testen
- Testkonzepte „Einmal auf dem Mainframe aufnehmen, mehrfach in AWS wiedergeben“
- IaC-Erstellung von Zielumgebungen mithilfe von vom Benutzer bereitgestellten Vorlagen CloudFormation

- Hohe Wiederholbarkeit der Tests
- Entwickelt für die Cloud, mit Blick auf Skalierbarkeit und Elastizität
- Tests in großem Maßstab mit hohem Automatisierungsgrad
- Kosteneffizienz

## Integration mit AWS CloudFormation

Anwendungstests verwenden Infrastruktur als Code mit AWS CloudFormation. Diese Designauswahl vereinfacht und verbessert Ihr Testerlebnis. AWS CloudFormation gibt Ihnen Autonomie und die Unabhängigkeit, die bessere Infrastruktur für Ihre Bedürfnisse zu definieren. Sie können viele Parameter (Instanzgröße, RDS-Instanz, optimale Sicherheitsgruppe) unabhängig voneinander auswählen oder definieren. Sie können Ressourcen hinzufügen, z. B. eine Amazon SQS SQS-Warteschlange, die Sie benötigen, damit Ihre Anwendung unter Testbedingungen ordnungsgemäß funktioniert.

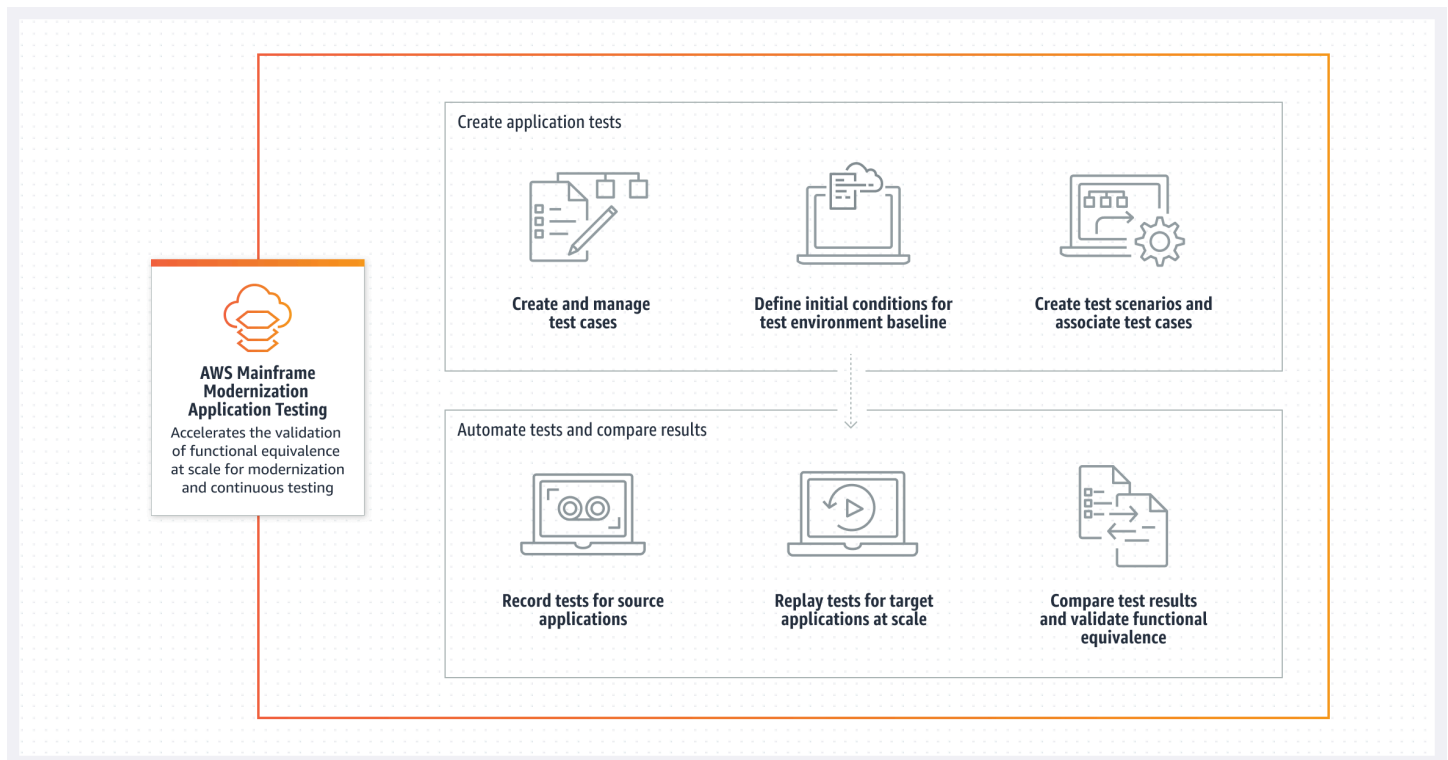
In den AWS CloudFormation Vorlagen, die zum Herunterladen bereitgestellt werden, werden Sie einige allgemeine Funktionen feststellen:

- Application Testing erstellt einen vollständig isolierten Stack, der eine Laufzeitumgebung und eine Anwendung für die AWS Mainframe-Modernisierung mit eigenen Netzwerk- und Sicherheitsdefinitionen umfasst. Dieser isolierte Stack bietet Ausfallsicherheit, da andere Akteure in demselben Stack die AWS-Konto Testaktivitäten nicht beeinträchtigen können. Außerdem werden Situationen vermieden, in denen Systembetreiber die Standard-VPC oder Sicherheitsgruppe ändern, was zu Fehlern bei der Testaktivität führen kann.
- Mit der Sicherheitsgruppe können Sie auch den externen Zugriff auf die beim Testen verwendeten Ressourcen kontrollieren. Eine Datenbank kann beispielsweise vertrauliche Daten enthalten.
- Eine vollständige Isolierung verhindert, dass andere Akteure, die sich die VPC teilen, den Datenverkehr ausspionieren.
- Es verbessert die Leistung. Beispielsweise erfolgt die Kommunikation zwischen der AWS Mainframe-Modernisierungsanwendung, die die Vorlage erstellt, und ihrer Amazon RDS-Datenbank in einem separaten Netzwerk (einer privaten VPC), wodurch vermieden wird, dass andere Akteure den Verkehr verlangsamen.

Wir empfehlen, dass Sie diese Funktionen auch in den von Ihnen AWS CloudFormation erstellten Vorlagen implementieren.

## Wie funktioniert das Testen von Anwendungen

Die folgende Abbildung gibt einen Überblick darüber, wie funktionale Äquivalenztests in Application Testing funktionieren.



- - Sie können Eingabedaten von der Quelle an AWS Mainframe Modernization [File Transfer](#) oder Ihre bevorzugten Tools für die Mainframe-Datenübertragung übertragen.
- Sie führen dieselbe Geschäftslogik sowohl auf der Quelle als auch auf dem Ziel aus.
- Application Testing vergleicht automatisch die Ausgabedaten (Datensätze, relationale Datenbankänderungen, 3270 Bildschirme und Benutzerinteraktionen) von Quelle und Ziel. Nachdem Sie Ihr Testszenario auf dem Mainframe ausgeführt haben, erfassen Sie die Ausgabedaten und übertragen sie an das Ziel. Anschließend spielen Sie das Testszenario auf AWS dem Ziel erneut ab. Application Testing vergleicht automatisch die Ausgabedaten des Testlaufs AWS mit den Ausgabedaten aus der Quelle. Sie können auf einen Blick sehen, welche Datensätze identisch, gleichwertig, unterschiedlich sind oder fehlen. Darüber hinaus können Sie Äquivalenzregeln definieren, sodass Datensätze, die nicht identisch sind, aber dieselbe geschäftliche Bedeutung haben, als gleichwertig angesehen werden.

Der Arbeitsablauf, dem Sie bei Application Testing folgen, besteht aus den folgenden Schritten:

1. Erstellen Sie Testfälle. Testfälle sind die kleinste Einheit von Testaktionen. Wenn Sie einen Testfall erstellen, identifizieren Sie auch die zu vergleichenden Datentypen, die die funktionale Äquivalenz zwischen Quelle und Ziel am besten darstellen.
2. Erstellen Sie Testszenarien. Testszenarien gruppieren verwandte Testfälle in einer bestimmten Reihenfolge zur Ausführung.
3. Erstellen Sie Anfangsbedingungen. In den Anfangsbedingungen wird erklärt, wie ein Test auf dem Mainframe aufgezeichnet und AWS CloudFormation anschließend automatisch derselbe Status erzeugt wird. AWS
4. Auf der Quelle aufnehmen und auf dem Ziel wiedergeben. Erfassen Sie die Eingabe- und Ausgabedatensätze auf dem Mainframe und laden Sie sie auf hoch. AWS Spielen Sie das Testszenario anschließend erneut ab. AWS
5. Vergleichen Sie Quell- und Zieldatensätze. Application Testing vergleicht automatisch die Ausgabedatensätze von Quell- und Zieldatensätzen, sodass Sie auf einen Blick erkennen können, was richtig ist und was nicht.

Sowohl die letzte Aktion eines Testszenarios als auch das Ziel des gesamten Prozesses besteht darin, Diskrepanzen zwischen den Quell- und den Zieltestläufen zu identifizieren. Application Testing vergleicht die Quellversion und die Zielversion für die Daten, die während des Testlaufs auf allen Interaktionskanälen erfasst wurden. Außerdem werden die Endzustände der relevanten Daten (wie in den Testfällen definiert) verglichen.

## Zugehörige Services

Anwendungstests sind eine Funktion der AWS Mainframe-Modernisierung. Außerdem wird Infrastruktur als Code verwendet AWS CloudFormation , um die Wiederholbarkeit, Automatisierung und Kosteneffizienz der Tests sicherzustellen. Weitere Informationen finden Sie hier:

- [AWS Mainframe-Modernisierung](#)
- [AWS CloudFormation](#)

## Zugriff auf Anwendungstests

Sie können von der AWS Mainframe Modernization Console aus auf Application Testing zugreifen, indem Sie in der linken Navigationsleiste Application Testing wählen.

## Preise für Anwendungstests

Die Preise für Anwendungstests finden Sie unter [AWS Mainframe Modernization Pricing](#).

## AWS Konzepte für das Testen von Anwendungen zur Mainframe-Modernisierung

AWS Application Testing ist als Vorabversion für AWS Mainframe Modernization verfügbar und kann sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

AWS Application Testing verwendet Begriffe, die andere Testdienste oder Softwarepakete möglicherweise verwenden, mit einer etwas anderen Bedeutung. In den folgenden Abschnitten wird erklärt, wie AWS Mainframe Modernization Application Testing diese Terminologie verwendet.

### Themen

- [Testfall](#)
- [Testszenario](#)
- [Projekt testen](#)
- [Ursprünglicher Zustand](#)
- [Aufzeichnen \(Erfassen\)](#)
- [Erneut abspielen](#)
- [Compare](#)
- [Datenbankvergleiche](#)
- [Vergleiche von Datensätzen](#)
- [Status des Vergleichs](#)
- [Regeln für die Gleichwertigkeit](#)
- [Vergleich von Datensätzen im Endzustand](#)
- [Vergleiche zwischen Datenbanken zum Stand und zum Fortschritt](#)
- [Funktionale Äquivalenz \(FE\)](#)
- [Online-Bildschirmvergleiche von 3270](#)



- [Aufnahme läuft](#)
- [Daten erneut abspielen](#)
- [Referenzdaten](#)
- [Aufnahmen, Wiedergeben und Vergleichen](#)
- [Unterschiede](#)
- [Äquivalenzen](#)
- [Quellanwendung](#)
- [Zielanwendung](#)

## Testfall

Ein Testfall ist die individuellste Aktionseinheit in Ihrem Test-Workflow. Normalerweise wird ein Testfall verwendet, um eine unabhängige Einheit der Geschäftslogik darzustellen, die Daten modifiziert. Vergleiche werden für jeden Testfall durchgeführt. Testfälle werden einem Testszenario hinzugefügt. Testfälle enthalten Metadaten zu den Datenartefakten (Datensätze, Datenbanken), die durch den Testfall geändert werden, und zu den Geschäftsfunktionen, die während der Testfallausführung ausgelöst werden: Batch-Jobs, 3270 interaktive Dialoge und andere. Zum Beispiel die Namen und Codepages von Datensätzen.

Eingabedaten → Testfall → Ausgabedaten

Testfälle können entweder online oder stapelweise sein:

- Online-Testfälle sind Testfälle, bei denen der Benutzer interaktive Bildschirmdialoge (3270) ausführt, um neue Geschäftsdaten (Datenbank und/oder Datensatzdatensätze) zu lesen, zu ändern oder zu erstellen.
- Batch-Testfälle sind Testfälle, bei denen ein Batch eingereicht werden muss, um neue Geschäftsdaten (Datensätze und/oder Datenbankdatensätze) zu lesen, zu verarbeiten und zu ändern oder zu erzeugen.

## Testszenario

Testszenarien sind eine Reihe von Testfällen, die nacheinander ausgeführt werden. Die Wiedergabe erfolgt auf der Ebene eines Testszenarios. Alle Testfälle im Testszenario werden in der Zieltestumgebung ausgeführt, wenn ein Testszenario wiederholt wird. Wenn es nach dem Vergleich

von Referenz- und Wiederholungstestartefakten Unterschiede gibt, werden die Unterschiede auf Testfallebene angezeigt.

Zum Beispiel Testszenario A:

Testfall 1, Testfall 2, Testfall 3 usw.

## Projekt testen

Testprojekte stellen eine Sammlung von Testszenarien dar, um einen gewünschten Testmeilenstein zu erreichen. Beispielsweise kann die Migration einer bestimmten Anwendung als ein einzelnes Testprojekt betrachtet werden. Durch die Gruppierung von Testszenarien in Testprojekte können Testmanager den Status von Testprojekten verfolgen, einschließlich bestandener/fehlgeschlagener Tests.

## Ursprünglicher Zustand

Eine Anfangsbedingung umfasst eine Reihe von Ressourcen (Rechenleistung, Datenspeicher usw.), die Sie erstellen müssen, sowie Anwendungsdaten, die Sie auf diesen erstellten Ressourcen wiederherstellen müssen, bevor Sie Testszenarien ausführen können. Dadurch wird die Ziel-Baseline der Testumgebung erstellt. Es ermöglicht Ihnen, eine AWS CloudFormation Vorlage bereitzustellen. Sie verwenden die Vorlage, um die Zieltestumgebung und optional den DDL-Extrakt aus der Quelldatenbank zu erstellen, wenn Ihr Testszenario Datenbankeinträge ändert. Jedes Testszenario wird einer Ausgangsbedingung zugeordnet. Eine Anfangsbedingung kann mehreren Testszenarien zugeordnet werden. Um Wiederholbarkeit und Konsistenz der Ergebnisse sicherzustellen und Fehlalarme aufgrund bereits geänderter Daten zu vermeiden, müssen Sie vor jedem Testszenario die Anfangsbedingungen wiederherstellen.

Bei Testszenarien, die Testfälle enthalten, bei denen Datenbankeinträge geändert werden, verweist die Anfangsbedingung auch auf einen DDL-Export der Quelldatenbankschemas und -tabellen.

## Aufzeichnen (Erfassen)

Aufzeichnungen werden auf der Ebene eines Testszenarios erstellt. Während der Aufzeichnung müssen Sie einen Amazon S3 S3-Speicherort angeben, der die Artefakte, Datensätze und CDC-Journale der relationalen Datenbank vom Quell-Mainframe enthält, mit denen verglichen werden soll. Diese werden als Referenzdaten aus dem Quell-Mainframe betrachtet. Während der Wiedergabe werden die generierten Wiedergabedaten mit den aufgezeichneten Referenzdaten verglichen, um die Gleichwertigkeit der Anwendung sicherzustellen.

## Erneut abspielen

Wiederholungen werden auf der Ebene eines Testszenarios durchgeführt. Während der Wiedergabe verwendet AWS Mainframe Modernization Application Testing das CloudFormation Skript, auf das in der zugehörigen Anfangsbedingung verwiesen wird, um die Zieltestumgebung zu erstellen und die Anwendung auszuführen. Datensätze und Datenbankdatensätze, die während der Wiedergabe geändert werden, werden erfasst und mit den Referenzdaten vom Mainframe verglichen. In der Regel werden Sie einmal auf dem Mainframe aufnehmen und dann mehrmals wiedergeben, bis die funktionale Äquivalenz erreicht ist.

## Compare

Vergleiche werden automatisch durchgeführt, nachdem eine Wiederholung erfolgreich abgeschlossen wurde. Bei Vergleichen werden die referenzierten Daten, die Sie während der Aufnahmephase hochgeladen und erfasst haben, mit den während der Wiedergabephase generierten Wiedergabedaten verglichen. Vergleiche werden auf der Ebene einzelner Testfälle für Datensätze, Datenbankdatensätze und Online-Bildschirme separat durchgeführt.

## Datenbankvergleiche

Application Testing verwendet beim Vergleich von Änderungen in Datenbankdatensätzen zwischen der Quell- und der Zielanwendung eine Funktion zum Abgleich des Status und des Fortschritts. Beim State-Progress-Abgleich werden die Unterschiede in den einzelnen INSERT-, UPDATE- und DELETE-Anweisungen verglichen, im Gegensatz zum Vergleich von Tabellenzeilen am Ende des Prozesses. Der State-Progress-Abgleich ist effizienter als Alternativen und ermöglicht schnellere und genauere Vergleiche, indem nur geänderte Daten verglichen und Fehler im Transaktionsablauf erkannt werden, die sich selbst korrigieren. Mithilfe der CDC-Technologie (Changed Data Capture) kann Application Testing einzelne Änderungen in der Relationsdatenbank erkennen und sie zwischen Quelle und Ziel vergleichen.

Änderungen an der Relationsdatenbank werden an Quelle und Ziel durch den getesteten Anwendungscode mithilfe von DML-Anweisungen (Data Modification Language) wie SQL INSERT, UPDATE oder DELETE generiert, aber auch indirekt, wenn die Anwendung gespeicherte Prozeduren verwendet oder wenn Datenbank-Trigger für einige Tabellen gesetzt sind oder wenn CASCADE DELETE verwendet wird, um die referenzielle Integrität zu gewährleisten, wodurch automatisch zusätzliche Löschungen ausgelöst werden.

## Vergleiche von Datensätzen

Application Testing vergleicht automatisch die Referenz- und Wiedergabedatensätze, die auf den Quell- (Aufnahme-) und Zielsystemen (Wiedergabe) erzeugt wurden.

Um Datensätze zu vergleichen:

1. Beginnen Sie mit denselben Eingabedaten (Datensätze, Datenbank) sowohl auf der Quelle als auch auf dem Ziel.
2. Führen Sie Ihre Testfälle auf dem Quellsystem (Mainframe) aus.
3. Erfassen Sie die produzierten Datensätze und laden Sie sie in einen Amazon S3 S3-Bucket hoch. Sie können Eingabedatensätze von der Quelle an AWS CDC-Journale, Bildschirme und Datensätze übertragen.
4. Geben Sie den Speicherort des Amazon S3 S3-Buckets an, in den die Mainframe-Datensätze hochgeladen wurden, als Sie den Testfall aufgezeichnet haben.

Nach Abschluss der Wiedergabe vergleicht Application Testing automatisch die ausgegebenen Referenz- und Zieldatensätze und zeigt, ob die Datensätze identisch, gleichwertig, unterschiedlich sind oder fehlen. Beispielsweise werden Datumsfelder, die sich auf den Zeitpunkt der Workload-Ausführung beziehen (Tag +1, Ende des aktuellen Monats usw.), automatisch als gleichwertig betrachtet. Darüber hinaus können Sie optional Äquivalenzregeln definieren, sodass Datensätze, die nicht identisch sind, immer noch dieselbe geschäftliche Bedeutung haben und als gleichwertig gekennzeichnet werden.

## Status des Vergleichs

Application Testing verwendet die folgenden Vergleichsstatus: IDENTISCH, ÄQUIVALENT und UNTERSCHIEDLICH.

### IDENTISCH

Die Quell- und Zieldaten sind exakt identisch.

### ÄQUIVALENT

Die Quell- und Zieldaten enthalten falsche Unterschiede, die als Äquivalenzen angesehen werden, z. B. Datums- oder Zeitstempel, die sich nicht auf die funktionale Gleichwertigkeit auswirken, wenn sie sich auf den Zeitpunkt der Workload-Ausführung beziehen. Sie können

Äquivalenzregeln definieren, um zu ermitteln, worin diese Unterschiede bestehen. Wenn alle wiedergegebenen Testszenarien im Vergleich zu ihren Referenztestszenarien den Status IDENTISCH oder ÄQUIVALENT aufweisen, beweist Ihr Testszenario die funktionale Gleichwertigkeit.

## ANDERS

Die Quell- und Zieldaten enthalten Unterschiede, z. B. eine unterschiedliche Anzahl von Datensätzen in einer Datenmenge oder unterschiedliche Werte in demselben Datensatz.

## Regeln für die Gleichwertigkeit

Eine Reihe von Regeln zur Identifizierung falscher Unterschiede, die als gleichwertige Ergebnisse angesehen werden können. Offline-Funktionale Äquivalenztests (OFET) führen bei einigen Ergebnissen unweigerlich zu Unterschieden zwischen dem Quell- und dem Zielsystem. Beispielsweise sind die Zeitstempel für Aktualisierungen konstruktionsbedingt unterschiedlich. Die Äquivalenzregeln erläutern, wie diese Unterschiede ausgeglichen und Fehlalarme beim Vergleich vermieden werden können. Wenn ein Datum beispielsweise Laufzeit + 2 Tage in einer bestimmten Datenspalte ist, beschreibt es die Äquivalenzregel und akzeptiert eine Zeit auf dem Zielsystem, die Laufzeit am Zielwert + 2 Tage ist, anstatt einen Wert, der genau derselben Spalte in der Referenzaufzeichnung entspricht.

## Vergleich von Datensätzen im Endzustand

Der Endstatus von Datensätzen, die erstellt oder geändert wurden, einschließlich aller Änderungen oder Aktualisierungen, die an den Datensätzen gegenüber ihrem ursprünglichen Zustand vorgenommen wurden. Bei Datensätzen untersucht Application Testing am Ende eines Testfalllaufs die Datensätze in diesen Datensätzen und vergleicht die Ergebnisse.

## Vergleiche zwischen Datenbanken zum Stand und zum Fortschritt

Vergleiche von Änderungen, die an Datenbankeinträgen vorgenommen wurden, als Folge einzelner DML-Anweisungen (Delete, Update, Insert). Application Testing vergleicht einzelne Änderungen (Einfügen, Aktualisieren oder Löschen einer Tabellenzeile) in der Quelldatenbank mit der Zieldatenbank und ermittelt Unterschiede für jede einzelne Änderung. Beispielsweise kann eine einzelne INSERT-Anweisung verwendet werden, um in eine Tabelle eine Zeile mit anderen Werten in der Quelldatenbank als in der Zieldatenbank einzufügen.

## Funktionale Äquivalenz (FE)

Zwei Systeme gelten als funktionell gleichwertig, wenn sie bei gleichen Eingabedaten bei allen beobachtbaren Operationen dieselben Ergebnisse liefern. Zwei Anwendungen gelten beispielsweise als funktionell gleichwertig, wenn dieselben Eingabedaten identische Ausgabedaten erzeugen (durch Bildschirme, Datensatzänderungen oder Datenbankänderungen).

### Online-Bildschirmvergleiche von 3270

Vergleicht die Ausgabe der Mainframe-3270-Bildschirme mit der Ausgabe der Webbildschirme der modernisierten Anwendung, wenn das Zielsystem unter AWS Blu Age-Laufzeit in der läuft. AWS Cloud Und es vergleicht die Ausgabe der Mainframe-3270-Bildschirme mit den 3270-Bildschirmen der neu gehosteten Anwendung, wenn das Zielsystem unter Micro Focus Runtime in der läuft. AWS Cloud

### Aufnahme läuft

Der Vorgang, bei dem ein bekannter Datenstatus wiederhergestellt und anschließend Referenzdaten eines Referenztestszenarios (für einen oder mehrere Testfälle nacheinander) auf einem Quellsystem erfasst oder aufgezeichnet werden.

### Daten erneut abspielen

Wiedergabedaten werden verwendet, um die Daten zu beschreiben, die durch die Wiedergabe eines Testszenarios in der Zieltestumgebung generiert wurden. Beispielsweise werden Wiedergabedaten generiert, wenn ein Testszenario auf einer AWS Mainframe-Modernisierungsdienstanwendung ausgeführt wird. Die Wiedergabedaten werden dann mit den Referenzdaten verglichen, die von der Quelle erfasst wurden. Jedes Mal, wenn Sie den Workload in der Zielumgebung wiedergeben, wird eine neue Generation von Wiedergabedaten generiert.

### Referenzdaten

Referenzdaten werden verwendet, um die auf dem Quell-Mainframe erfassten Daten zu beschreiben. Dies ist die Referenz, mit der die von der Wiedergabe (Ziel) generierten Daten verglichen werden. Normalerweise gibt es für jeden Datensatz auf dem Mainframe, der Referenzdaten erstellt, viele Wiederholungen. Das liegt daran, dass Benutzer in der Regel den korrekten Status der Anwendung auf dem Mainframe erfassen und die Testfälle auf der modernisierten Zielanwendung erneut abspielen, um die Gleichwertigkeit zu überprüfen. Wenn Fehler gefunden werden, werden sie

behalten und die Testfälle werden erneut abgespielt. Oft werden mehrere Zyklen wiederholt, Fehler behoben und erneut abgespielt, um das Auftreten zu überprüfen. Dies wird als Testparadigma bezeichnet: einmal aufzeichnen, mehrfach wiedergeben.

## Aufnehmen, Wiedergeben und Vergleichen

Das Testen von Anwendungen erfolgt in drei Schritten:

- **Datensatz:** erfasst die referenzierten Daten, die auf dem Mainframe für jeden Testfall eines Testszenarios erstellt wurden. Dazu können 3270 Online-Bildschirme, Datensätze und Datenbankdatensätze gehören.
  - Für Online-3270-Bildschirme müssen Sie den Blu Insights-Terminalemulator verwenden, um Ihren Quell-Workload zu erfassen. Weitere Informationen finden Sie in der [Blu Insights-Dokumentation](#).
  - Bei Datensätzen müssen Sie die von jedem Testfall auf dem Mainframe erzeugten Datensätze mithilfe gängiger Tools wie FTP oder dem Dataset-Transfer-Service, der Teil von AWS Mainframe Modernization ist, erfassen.
  - Für Datenbankänderungen verwenden Sie die [AWS Mainframe Modernization Data Replication with Precisely](#) Dokumentation, um CDC-Journale mit Änderungen zu erfassen und zu generieren.
- **Wiederholung:** Das Testszenario wird in der Zielumgebung wiedergegeben. Alle im Testszenario angegebenen Testfälle werden ausgeführt. Spezifizierte Datentypen, die durch die einzelnen Testfälle erstellt wurden, wie Datensätze, Änderungen an relationalen Datenbanken oder 3270-Bildschirme, werden automatisiert erfasst. Diese Daten werden als Wiederholungsdaten bezeichnet und mit den während der Aufzeichnungsphase erfassten Referenzdaten verglichen.

### Note

Für die Änderungen an der relationalen Datenbank sind DMS-spezifische Konfigurationsoptionen in Ihrer Vorlage für Ausgangsbedingungen erforderlich.  
CloudFormation

- **Vergleich:** Die Referenzdaten für den Quelltest und die Ziel-Wiedergabedaten werden verglichen, und die Ergebnisse werden Ihnen als identische, unterschiedliche, gleichwertige oder fehlende Daten angezeigt.

## Unterschiede

Zeigt an, dass beim Datenvergleich Unterschiede zwischen den Referenz- und Wiedergabedatensätzen festgestellt wurden. Beispielsweise wird ein Feld auf einem Online-3270-Bildschirm, das aus Sicht der Geschäftslogik unterschiedliche Werte zwischen dem Quell-Mainframe und der modernisierten Ziellanwendung anzeigt, als Unterschied betrachtet. Ein anderes Beispiel ist ein Datensatz in einem Datensatz, der zwischen Quell- und Ziellanwendungen nicht identisch ist.

## Äquivalenzen

Äquivalente Datensätze sind Datensätze, die sich zwischen den Referenz- und Wiedergabedatensätzen unterscheiden, die aber aus Sicht der Geschäftslogik nicht als unterschiedlich behandelt werden sollten. Zum Beispiel ein Datensatz, der den Zeitstempel enthält, wann der Datensatz erstellt wurde (Workload-Ausführungszeit). Mithilfe anpassbarer Äquivalenzregeln können Sie Application Testing anweisen, eine solche falsch positive Differenz als Äquivalenz zu behandeln, auch wenn sie unterschiedliche Werte zwischen Referenz- und Wiedergabedaten anzeigt.

## Quellanwendung

Die Mainframe-Quellanwendung, mit der verglichen werden soll.

## Ziellanwendung

Die neue oder geänderte Anwendung, auf der getestet wird und die mit der Quellanwendung verglichen wird, um etwaige Fehler zu erkennen und die funktionale Äquivalenz zwischen Quell- und Ziellanwendung zu erreichen. Die Ziellanwendung wird normalerweise in der AWS Cloud ausgeführt.

## Tutorial: Richten Sie die CardDemo Beispielanwendung ein

AWS Application Testing befindet sich in der Vorschauversion für AWS Mainframe Modernization und kann sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

In diesem Tutorial erstellen Sie einen AWS CloudFormation Stack, der Ihnen bei der Einrichtung der [CardDemo Beispielanwendung](#) für die Neuplattformierung mit Micro Focus on AWS Mainframe Modernization Managed Service und Funktionen wie AWS Mainframe Modernization



Application Testing hilft. Das Tutorial beschreibt eine AWS CloudFormation Beispielvorgabe, mit der Sie den Stack erstellen können. Wir stellen auch eine komprimierte Datei mit den erforderlichen Anwendungsartefakten zur Verfügung. Die Beispielvorgabe stellt eine Datenbank, eine Laufzeitumgebung, eine Anwendung und eine vollständig isolierte Netzwerkumgebung bereit.

Diese Vorlage erstellt mehrere AWS Ressourcen. Sie werden Ihnen in Rechnung gestellt, wenn Sie anhand dieser Vorlage einen Stapel erstellen.

## Voraussetzungen

- Laden Sie das und entpacken Sie es. [IC3-card-demo-zip](#) [datasets\\_Mainframe\\_ebcdic.zip](#) Diese Dateien enthalten das CardDemo Beispiel und die Beispieldatensätze zur Verwendung mit AWS Anwendungstests.
- Erstellen Sie einen Amazon S3 S3-Bucket für die CardDemo Dateien und andere Artefakte. z. B. `my-carddemo-bucket`.

## Schritt 1: Bereiten Sie die Einrichtung vor CardDemo

Laden Sie die CardDemo Beispieldateien hoch und bearbeiten Sie die AWS CloudFormation Vorlage, mit der die CardDemo Anwendung erstellt wird.

1. Laden Sie die IC3-card-demo Ordner `datasets_Mainframe_ebcdic` und, die Sie zuvor entpackt haben, in Ihren Bucket hoch.
2. Laden Sie die `aws-m2-math-mf-carddemo.yaml` AWS CloudFormation Vorlage aus Ihrem Bucket herunter. Sie befindet sich im IC3-card-demo Ordner.
3. Bearbeiten Sie die `aws-m2-math-mf-carddemo.yaml` AWS CloudFormation Vorlage wie folgt:
  - Ändern Sie den `BucketName` Parameter in den Namen des Buckets, den Sie zuvor definiert haben, z. `my-carddemo-bucket` B.
  - Ändern Sie `ImportJsonPath` den Speicherort der `mf-carddemo-datasets-import.json` Datei in Ihrem Bucket. `s3://my-carddemo-bucket/IC3-card-demo/mf-carddemo-datasets-import.json` Wenn Sie beispielsweise diesen Wert aktualisieren, wird sichergestellt, dass die Ausgabe den richtigen Wert `M2ImportJson` hat.
  - (Optional) Passen Sie die `InstanceType` Parameter `EngineVersion` und an Ihre Standards an.

**Note**

Ändern Sie die `M2ApplicationId` Ausgaben `M2EnvironmentId` und nicht. Application Testing verwendet diese Werte, um die Ressourcen zu ermitteln, mit denen es interagieren wird.

## Schritt 2: Erstellen Sie alle erforderlichen Ressourcen

Führen Sie Ihre benutzerdefinierte AWS CloudFormation Vorlage aus, um alle Ressourcen zu erstellen, die Sie benötigen, um dieses Tutorial erfolgreich abzuschließen. Diese Vorlage richtet die CardDemo Anwendung so ein, dass Sie sie beim Testen verwenden können.

1. Melden Sie sich bei der AWS CloudFormation Konsole an und wählen Sie Stack erstellen und anschließend Mit neuen Ressourcen (Standard).
2. Wählen Sie unter Voraussetzung — Vorlage vorbereiten die Option Vorlage ist bereit aus.
3. Wählen Sie unter Vorlage angeben die Option Vorlagendatei hochladen und anschließend Datei auswählen aus.
4. Navigieren Sie zu dem Ort, an dem Sie die Datei heruntergeladen haben, wählen Sie sie aus `aws-m2-math-mf-carddemo.yaml` und klicken Sie dann auf Weiter.
5. Geben Sie unter Stackdetails angeben einen Namen für den Stapel ein, sodass Sie ihn leicht in einer Liste finden können, und wählen Sie dann Weiter aus.
6. Behalten Sie unter Stack-Optionen konfigurieren die Standardwerte bei und wählen Sie Weiter aus.
7. Prüfen Sie unter Überprüfen, was AWS CloudFormation für Sie erstellt wurde, und wählen Sie dann Absenden aus.

Die Erstellung des Stacks dauert etwa 10—15 Minuten AWS CloudFormation .

**Note**

Die Vorlage ist so eingerichtet, dass sie den Namen der Ressourcen, die sie erstellt, ein eindeutiges Suffix anhängt. Das bedeutet, dass Sie mehrere Instanzen dieser Stack-Vorlage

parallel erstellen können. Dies ist eine wichtige Funktion für Anwendungstests, mit der Sie mehrere Testszenarien gleichzeitig ausführen können.

## Schritt 3: Stellen Sie die Anwendung bereit und starten Sie sie

Stellen Sie die CardDemo Anwendung bereit, die für Sie AWS CloudFormation erstellt wurde, und stellen Sie sicher, dass sie ausgeführt wird.

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole und wählen Sie im linken Navigationsbereich Anwendungen aus.
2. Wählen Sie die CardDemo Anwendung aus, die etwa so `aws-m2-math-mf-carddemo-abc1d2e3` heißt.
3. Wählen Sie „Aktionen“ und anschließend „Anwendung bereitstellen“.
4. Wählen Sie unter Umgebungen die Laufzeitumgebung aus, die der Anwendung entspricht. An das Ende des Namens wird dieselbe eindeutige Kennung angehängt. z. B. `aws-m2-math-mf-carddemo-abc1d2e3`.
5. Wählen Sie Bereitstellen. Warten Sie, bis die Anwendung erfolgreich bereitgestellt wurde und sich im Status Bereit befindet.
6. Wählen Sie die Anwendung aus, klicken Sie dann auf Aktionen und dann auf Anwendung starten. Warten Sie, bis sich die Anwendung im Status Running befindet.
7. Kopieren Sie auf der Seite mit den Anwendungsdetails den Port und den DNS-Hostnamen, die Sie benötigen, um eine Verbindung mit der laufenden Anwendung herzustellen.

## Schritt 4: Erste Daten importieren

Um die CardDemo Beispielanwendung verwenden zu können, müssen Sie einen ersten Datensatz importieren. Führen Sie folgende Schritte aus.

1. Laden Sie die Datei `mf-carddemo-datasets-import.json` herunter.
2. Bearbeiten Sie die Datei in Ihrem bevorzugten Texteditor.
3. Suchen Sie den `s3Location` Parameter und aktualisieren Sie den Wert so, dass er auf den Amazon S3-Bucket verweist, den Sie erstellt haben.
4. Nehmen Sie dieselbe Änderung für jedes Vorkommen von `s3Location` und speichern Sie dann die Datei.

5. Melden Sie sich bei der Amazon S3 S3-Konsole an und navigieren Sie zu dem Bucket, den Sie zuvor erstellt haben.
6. Laden Sie die benutzerdefinierte `mf-carddemo-datasets-import.json` Datei hoch.
7. Öffnen Sie die AWS Mainframe-Modernisierungskonsole und wählen Sie im linken Navigationsbereich Anwendungen aus.
8. Wählen Sie die CardDemo Anwendung aus.
9. Wählen Sie Datensätze und dann Import.
10. Navigieren Sie zu dem Speicherort in Amazon S3, in den Sie die benutzerdefinierte JSON-Datei hochgeladen haben, und wählen Sie Senden.

Dieser Job importiert 23 Datensätze. Um das Ergebnis des Importjobs zu überwachen, überprüfen Sie die Konsole. Wenn alle Datensätze erfolgreich importiert wurden, stellen Sie eine Verbindung zur Anwendung her.

#### Note

Wenn Sie diese Vorlage in Anwendungstests verwenden, übernimmt Output `M2ImportJson` automatisch den Importvorgang.

## Schritt 5: Connect zur CardDemo Anwendung her

Stellen Sie mit dem 3270-Emulator Ihrer Wahl eine Connect zur CardDemo Beispielanwendung her.

- Wenn die Anwendung ausgeführt wird, verwenden Sie Ihren 3270-Emulator, um eine Verbindung zur Anwendung herzustellen, und geben Sie gegebenenfalls den DNS-Hostnamen und den Portnamen an.

Wenn Sie beispielsweise den [Open-Source-Emulator c3270](#) verwenden, sieht Ihr Befehl wie folgt aus:

```
c3270 -port port-number DNS-hostname
```

## port

Der auf der Anwendungsdetailseite angegebene Port. Zum Beispiel 6000.

## Hostname

Der auf der Anwendungsdetailseite angegebene DNS-Hostname.

Die folgende Abbildung zeigt, wo der Port und der DSN-Hostname zu finden sind.

The screenshot displays the AWS Mainframe Modernization console interface for an application named 'aws-m2-math-mf-carddemo-7f28a650'. The 'Application information' section is expanded, showing various details. Two red arrows point to specific fields: one points to the 'Ports' field, which contains the value '7000', and the other points to the 'DNS Hostname' field, which contains the value 'haytgmjvgazteoi-ibgcq4di.m2.us-west-2.amazonaws.com'. Other visible fields include Name, Status (Running), ARN, Creation time, KMS key, Engine (Micro Focus), and Description.

Application information			
Name	Status	Ports	Logs
aws-m2-math-mf-carddemo-7f28a650	Running	7000	<a href="#">ConsoleLog</a> <a href="#">BatchJobLogs</a>
ARN	Creation time	KMS key	Description
arn:aws:m2:us-west-2:app/efzltb7ocfb5zi7fwfcxfusw4	May 2, 2023 at 10:50 (UTC-04:00)	AWS owned key	m2 application: aws-m2-math-mf-carddemo-7f28a650
Engine	DNS Hostname		
Micro Focus	haytgmjvgazteoi-ibgcq4di.m2.us-west-2.amazonaws.com		

## Tutorial: AWS Mainframe-Modernisierung, Anwendungstests, Wiederholung und Vergleich unter Verwendung von CardDemo AWS Blu Age, bereitgestellt auf Amazon EC2

AWS Anwendungstests befinden sich in der Vorschauversion für AWS Mainframe Modernization und können sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

In diesem Tutorial führen Sie die erforderlichen Schritte durch, um Test-Workloads mit der CardDemo Anwendung, die auf AWS Blu Age läuft und auf Amazon EC2 bereitgestellt wird, abzuspielen und zu vergleichen.

## Schritt 1: Besorgen Sie sich AWS Blu Age Amazon EC2 Amazon Machine Image (AMI)

Folgen Sie den Anweisungen im [Setup-Tutorial für AWSAWS Blu Age Runtime \(auf Amazon EC2\)](#) für die Onboarding-Schritte, die für den Zugriff auf AWS Blu Age on Amazon EC2 AMI erforderlich sind.

## Schritt 2: Starten Sie eine Amazon EC2 EC2-Instance mit dem AWS Blu Age AMI

1. Richten Sie Ihre AWS Anmeldedaten ein.
2. Identifizieren Sie den Speicherort der Amazon EC2 AMI-Binärdatei 3.5.0 (nur AWS CLI/Blu Age-Version) aus dem Amazon S3 S3-Bucket:

```
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/  
aws s3 ls s3://aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1/3.5.0/AMI/
```

### Note

Die Funktion für Anwendungstests kann nur in 4 Regionen in Prod verwendet werden (us-east-1, sa-east-1, eu-central-1 und ap-southeast-2).

3. Stellen Sie das AMI in Ihrem Konto mit dem folgenden Befehl wieder her:

```
aws ec2 create-restore-image-task --object-key 3.5.0/AMI/ami-0182ffe3b9d63925b.bin  
--bucket aws-bluage-runtime-artifacts-xxxxxxx-eu-west-1 --region eu-west-1 --name  
"AWS BLUAGE RUNTIME AMI"
```

### Note

Ersetzen Sie den Namen der AMI-Bin-Datei und die Region, in der Sie das AMI erstellen möchten.

4. Nachdem Sie eine Amazon EC2 EC2-Instance erstellt haben, finden Sie die richtige AMI-ID, mit der das AMI aus dem Amazon S3 S3-Bucket wiederhergestellt wurde, im Amazon EC2 EC2-Image-Katalog.

**Note**

In diesem Tutorial lautet die AMI-ID `ami-0d0fafcc636fd1e6d`, und Sie müssen diese ID in den verschiedenen Konfigurationsdateien in die ID ändern, die Ihnen zur Verfügung gestellt wurde.

1. Wenn `aws ec2 create-restore-image-task` fehlschlägt, überprüfen Sie Ihre Version von Python und CLI mit dem folgenden Befehl:

```
aws --version
```

**Note**

Die Python-Version muss  $\geq 3$  und die CLI-Version muss  $\geq 2$  sein.

2. Wenn diese Versionen veraltet sind, muss die CLI aktualisiert werden. Um die CLI zu aktualisieren:
  - a. Folgen Sie den Anweisungen unter [Installieren oder Aktualisieren der neuesten Version der AWS-CLI](#).
  - b. Entfernen Sie CLI v1 mit dem folgenden Befehl:

```
sudo yum remove awscli
```

- c. Und installiere CLI v2 mit den folgenden Befehlen:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install
```

- d. Überprüfen Sie abschließend die Version von Python und CLI mit dem folgenden Befehl:

```
aws --version
```

3. Sie können dann `aws create-restore-image-task ec2` wiederholen.

## Schritt 3: Laden Sie CardDemo abhängige Dateien auf S3 hoch

Kopieren Sie den Inhalt der Ordner Datenbanken, Dateisystem und Benutzerdaten. Laden Sie die Anwendungen herunter und entpacken Sie sie. CardDemo Diese drei Ordner müssen in einen Ihrer Buckets kopiert werden, der in dieser Dokumentation `your-s3-bucket` genannt wird.

## Schritt 4: Datenbanken laden und die Anwendung initialisieren CardDemo

Erstellen Sie eine temporäre Amazon EC2 EC2-Instance, die Sie als Rechenressource verwenden, um die erforderlichen Datenbank-Snapshots für die CardDemo Anwendung zu generieren. Diese EC2-Instance führt nicht die CardDemo Anwendung selbst aus, sondern generiert stattdessen die Datenbank-Snapshots, die später verwendet werden.

Bearbeiten Sie zunächst die bereitgestellte CloudFormation Vorlage mit dem Namen `'load-and-create-ba-snapshots.yml'` Dies ist die CloudFormation Vorlage, die verwendet wird, um die Amazon EC2 EC2-Instance zu erstellen, die zum Generieren der Datenbank-Snapshots verwendet wird.

1. Generieren Sie Ihr EC2-Schlüsselpaar, das für die EC2-Instance verwendet wird, und stellen Sie es bereit. Weitere Informationen finden Sie unter [Schlüsselpaare erstellen](#).

Beispiel:

```
Ec2KeyPair:
  Description: 'ec2 key pair'
  Default: 'm2-tests-us-west-2'
  Type: String
```

2. Geben Sie den Amazon S3 S3-Pfad Ihres Ordners an, in den Sie den Datenbankordner aus dem vorherigen Schritt abgelegt haben:

```
S3DBScriptsPath:
  Description: 'S3 DB scripts folder path'
  Type: String
  Default: 's3://your-s3-bucket/databases'
```

3. Geben Sie den Amazon S3 S3-Pfad Ihres Ordners an, in den Sie den Dateisystemordner aus dem vorherigen Schritt abgelegt haben:

```
S3ApplicationFilePath:
  Description: 'S3 application files folder path'
  Type: String
```



```
Default: 's3://your-s3-bucket/file-system'
```

4. Geben Sie den Amazon S3 S3-Pfad Ihres Ordners an, in den Sie den Benutzerdatenordner aus dem vorherigen Schritt abgelegt haben:

```
S3UserDataPath:  
  Description: 'S3 userdata folder path'  
  Type: String  
  Default: 's3://your-s3-bucket/userdata'
```

5. Geben Sie auch einen Amazon S3 S3-Pfad an, in dem Sie die Ergebnisdateien speichern, die im nächsten Schritt verwendet werden sollen.

```
S3SaveProducedFilePath:  
  Description: 'S3 path folder to save produced files'  
  Type: String  
  Default: 's3://your-s3-bucket/post-produced-files'
```

6. Ändern Sie die AMI-ID mithilfe der folgenden Vorlage durch die richtige, die Sie weiter oben in diesem Tutorial erhalten haben:

```
BaaAmiId:  
  Description: 'ami id (AL2) for ba anywhere'  
  Default: 'ami-0bd41245734fd20d9'  
  Type: String
```

- Sie können optional den Namen der drei Snapshots ändern, die beim Ausführen der Ladedatenbanken withCloudFormation erstellt werden. Diese sind während der Erstellung im CloudFormation Stapel sichtbar und werden später in diesem Tutorial verwendet. Denken Sie daran, sich die Namen zu notieren, die für die Datenbank-Snapshots verwendet wurden.

```
SnapshotPrimary:  
  Description: 'Snapshot Name DB BA Primary'  
  Type: String  
  Default: 'snapshot-primary'
```

```
SnapshotBluesam:  
  Description: 'Snapshot Name DB BA Bluesam'  
  Type: String  
  Default: 'snapshot-bluesam'
```

```
SnapshotJics:  
  Description: 'Snapshot Name DB BA Jics'  
  Type: String  
  Default: 'snapshot-jics'
```

**Note**

In diesem Dokument gehen wir davon aus, dass der Name der Snapshots einheitlich bleibt.

7. Führen Sie das CloudFormation mit der CLI oder der AWS Konsole aus, indem Sie die Schaltfläche „Stack erstellen“ und den Assistenten verwenden. Am Ende des Vorgangs sollten in der RDS-Konsole drei Snapshots mit dem von Ihnen ausgewählten Namen gefolgt von einer eindeutigen ID angezeigt werden. Sie benötigen diese Namen im nächsten Schritt.

**Note**

RDS fügt den in der Vorlage definierten Snapshot-Namen Postfixe hinzu. AWS CloudFormation Stellen Sie sicher, dass Sie den vollständigen Snapshot-Namen von RDS erhalten haben, bevor Sie mit dem nächsten Schritt fortfahren.

Beispiel für einen CLI-Befehl-

```
aws cloudformation create-stack --stack-name load-and-create-ba-snapshots --  
template-url https://your-apptest-bucket.s3.us-west-2.amazonaws.com/load-and-  
create-ba-snapshots.yml --capabilities CAPABILITY_NAMED_IAM
```

Sie können auch im Amazon S3 S3-Pfad, den Sie für S3 angegeben haben `SaveProducedFilePath`, überprüfen, ob die Datensätze korrekt erstellt wurden.

## Schritt 5: Starten Sie AWS Blu Age Runtime CloudFormation

Wird verwendet CloudFormation , um die Amazon EC2 EC2-Instance mit der CardDemo AWS Blu Age-Anwendung auszuführen. Sie müssen einige Variablen im CloudFormation Namen ersetzen, `m2-with-ba-using-snapshots-https-authentication.yml` indem Sie die YAML-Datei bearbeiten oder die Werte in der Konsole während des CFN-Starts ändern.

1. Ändern Sie den `AllowedVpcEndpointPrincipals`, um anzugeben, welches Konto den VPC-Endpunkt für den Zugriff auf die AWS Blu Age-Laufzeit erreicht. Verwenden Sie dazu die folgenden Befehle:

```
AllowedVpcEndpointPrincipals:  
  Description: 'comma-separated list of IAM users, IAM roles, or AWS accounts'  
  Default: 'apptest.amazonaws.com'  
  Type: String
```

2. Ändern Sie den Wert der Variablen `SnapshotPrimaryDb` und `SnapshotJicsDb` in den Namen der Snapshots. `SnapshotBlusamDb` Rufen Sie auch die Namen der Snapshots von RDS ab, nachdem sie im vorherigen Schritt erstellt wurden.

```
SnapshotPrimary:  
  Description: 'Snapshot DB cluster for DB Primary'  
  Type: String  
  Default: 'snapshot-primary87d067b0'
```

```
SnapshotBluesam:  
  Description: 'Snapshot DB cluster for DB Bluesam'  
  Type: String  
  Default: 'snapshot-bluesam87d067b0'
```

```
SnapshotJics:  
  Description: 'Snapshot DB cluster for DB Jics'  
  Type: String  
  Default: 'snapshot-jics87d067b0'
```

#### Note

RDS fügt den Snapshot-Namen ein eigenes Postfix hinzu.

3. Geben Sie Ihr Amazon EC2 EC2-Schlüsselpaar für die EC2-Instance mit diesem Befehl ein:

```
Ec2KeyPair:  
  Description: 'ec2 key pair'  
  Default: 'm2-tests-us-west-2'  
  Type: String
```

4. Geben Sie die AMI-ID ein, die Sie während des AMI-Registrierungsprozesses für die Variable erhalten haben `BaaAmild`, und verwenden Sie:

```
BaaAmiId:  
  Description: 'ami id (AL2) for ba anywhere'  
  Default: 'ami-0d0fafcc636fd1e6d'  
  Type: String
```

5. Geben Sie mit dem folgenden Befehl den Amazon S3 S3-Ordnerpfad an, den Sie im vorherigen Schritt zum Speichern der produzierten Dateien verwendet haben:

```
S3ApplicationFilePath:  
  Description: 'bucket name'  
  Type: String  
  Default: 's3://your-s3-bucket/post-produced-files'
```

6. Geben Sie abschließend den Ordnerpfad von s3- anuserdata-folder-path:

```
S3UserDataPath:  
  Description: 'S3 userdata folder path'  
  Type: String  
  Default: 's3://your-s3-bucket/userdata'
```

- (Optional) Sie können den HTTPS-Modus und die grundlegende HTTP-Authentifizierung für Tomcat aktivieren. Obwohl die Standardeinstellungen auch funktionieren würden.

#### Note

Standardmäßig ist der HTTPS-Modus deaktiviert und im Parameter auf den Modus HTTP gesetzt `BacHttpsMode`:

#### Beispielsweise:

```
BacHttpsMode:  
  Description: 'http or https for Blue Age Runtime connection mode '  
  Default: 'http'  
  Type: String  
  AllowedValues: [http, https]
```

- (Optional) Um den HTTPS-Modus zu aktivieren, müssen Sie den Wert in HTTPS ändern und den ARN Ihres ACM-Zertifikats angeben, indem Sie den Wert der Variablen CertArnACM ändern:

```
ACMCertArn:
  Type: String
  Description: 'ACM certificate ARN'
  Default: 'your arn certificate'
```

- (Optional) Die Standardauthentifizierung ist standardmäßig deaktiviert, und der Parameter ist auf false WithBacBasicAuthentication gesetzt. Sie können sie aktivieren, indem Sie den Wert auf true setzen.

```
WithBacBasicAuthentication:
  Description: 'false or true for Blue Age Runtime Basic Authentication '
  Default: false
  Type: String
  AllowedValues: [true, false]
```

7. Wenn Sie die Konfiguration abgeschlossen haben, können Sie den Stack mithilfe der bearbeiteten CloudFormation Vorlage erstellen.

## Schritt 6: Testen der AWS Blu Age Amazon EC2 EC2-Instance

Führen Sie die CloudFormation Vorlage manuell aus, um die AWS Blu Age Amazon EC2 EC2-Instance für die CardDemo Anwendung zu erstellen, um sicherzustellen, dass sie fehlerfrei gestartet wird. Auf diese Weise wird überprüft, ob die CloudFormation Vorlage und alle Voraussetzungen gültig sind, bevor die CloudFormation Vorlage mit der Funktion für Anwendungstests verwendet wird. Anschließend können Sie Application Testing verwenden, um die AWS Blu Age Amazon EC2 EC2-Zielinstanz während der Wiedergabe automatisch zu erstellen und eine Ausgangsbedingung zu vergleichen.

1. Führen CloudFormation Sie den Befehl create stack aus, um die AWS Blu Age Amazon EC2 EC2-Instance zu erstellen, und geben Sie dabei die CloudFormation Vorlage m2- with-ba-using-snapshots -https-authentication.yml an, die Sie im vorherigen Schritt bearbeitet haben:

```
aws cloudformation create-stack --stack-name load-and-create-ba-snapshots --
template-url https://apptest-ba-demo.s3.us-west-2.amazonaws.com/m2-with-ba-using-
```

```
snapshots-https-authentication.yml --capabilities CAPABILITY_NAMED_IAM --region us-west-2
```

**Note**

Denken Sie daran, die richtige Region anzugeben, in der das AWS Blu Age AMI wiederhergestellt wurde.

2. Stellen Sie sicher, dass alles ordnungsgemäß funktioniert, indem Sie in der Konsole nach der laufenden Amazon EC2 EC2-Instance suchen. Stellen Sie mit dem Sitzungsmanager eine Connect.
3. Nachdem Sie mit der Amazon EC2 EC2-Instance verbunden sind, verwenden Sie die folgenden Befehle:

```
sudo su  
cd /m2-anywhere/tomcat.gapwalk/velocity/logs  
cat catalina.log
```

4. Stellen Sie sicher, dass das Protokoll keine Ausnahmen oder Fehler enthält.
5. Überprüfen Sie als Nächstes, ob die Anwendung reagiert, indem Sie diesen Befehl verwenden:

```
curl http://localhost:8080/gapwalk-application/
```

Sie werden die Meldung „Die Jics-Anwendung läuft“ sehen.

## Schritt 7: Stellen Sie sicher, dass die vorherigen Schritte korrekt abgeschlossen wurden

In den nächsten Schritten werden wir AWS Mainframe Modernization Application Testing verwenden, um die von der Anwendung erstellten Datensätze erneut abzuspielen und zu vergleichen. CardDemo Diese Schritte setzen den erfolgreichen Abschluss aller vorherigen Schritte in diesem Tutorial voraus. Überprüfen Sie Folgendes, bevor Sie fortfahren:

1. Sie haben die AWS Blu Age on Amazon EC2 EC2-Instance erfolgreich mithilfe der AWS CloudFormation Vorlage erstellt.
2. Der Tomcat-Service auf dem AWS Blu Age auf Amazon EC2 ist ausnahmslos betriebsbereit.

Wenn Sie die EC2-Instance mit der CardDemo Anwendung zum Laufen gebracht haben, führen Sie die folgenden Schritte auf der Anwendungstestkonsole aus, um die Wiedergabe und den Vergleich von Batch-Datensätzen durchzuführen.

## Schritt 8. Erstellen Sie eine Ausgangsbedingung

In diesem Schritt erstellen Sie eine Ausgangsbedingung, indem Sie die CloudFormation Vorlage angeben, die Sie für die Bereitstellung der AWS Blu CardDemo Age-Anwendung auf Amazon EC2 verwendet haben.

1. [Öffnen Sie die AWS Mainframe Modernization Console unter https://console.aws.amazon.com/m2/](https://console.aws.amazon.com/m2/).
2. Wählen Sie im linken Navigationsbereich Application Testing aus.
3. Wählen Sie unter Anwendungstests die Option Anfangsbedingung erstellen aus.
4. Verwenden Sie Ihre lokale Kopie mit modifizierten Amazon S3 S3-Pfad- und Snapshot-IDs, die auf Ihre Ressourcen verweisen.

## Schritt 9: Erstellen Sie den Testfall

In diesem Schritt erstellen Sie den Testfall, der zum Vergleich der in der Card Demo-Anwendung erstellten Datensätze verwendet wird.

1. Erstellen Sie einen neuen Testfall. Geben Sie ihm einen Namen und eine Beschreibung.
2. Geben Sie CREAMTMT.JCL als JCL-Namen an.
3. Fügen Sie der Testfalldefinition die folgenden Datensätze hinzu:

Name	CCSID	RecordFormat	RecordLength
AWS.M2.CA RDDEMO.ST ATEMNT.PS	„037“	FB	80
AWS.M2.CA RDDEMO.ST ATEMNT.HTML	„037“	FB	100

**Note**

Ihr JCL-Name und die Datensatzdetails müssen übereinstimmen.

## Schritt 10: Erstellen Sie ein Testszenario

1. Erstellen Sie ein neues Testszenario und geben Sie einen Namen und eine Beschreibung dafür ein.
2. Fügen Sie den Testfall, den Sie im vorherigen Schritt erstellt haben, Ihrem Testszenario hinzu.
3. Sobald das Testszenario erstellt wurde, wählen Sie auf der Übersichtsseite des Testszenarios die in Schritt 1 erstellte Anfangsbedingung aus.

## Schritt 11: Nehmen Sie Ihr Testszenario auf

In diesem Schritt führen Sie Testfälle auf der Quelle aus. Um das zu tun:

1. Laden Sie die Datensätze herunter, die aus der Mainframe-Ausführung der Anwendung stammen, und führen Sie sie aus. CardDemo
2. Laden Sie den entpackten Ordner in Ihren Amazon S3 S3-Bucket hoch. Dieser Amazon S3 S3-Bucket muss sich in derselben Region befinden wie Ihre anderen Ressourcen für Anwendungstests.

**Note**

Es sollten zwei Dateien vorhanden sein, deren Namen den im vorherigen Testfall übergebenen Datensatznamen entsprechen.

3. Wählen Sie auf der Übersichtsseite für das Testszenario die Schaltfläche Aufzeichnen aus.
4. Geben Sie auf der Seite Testszenario-Datensatz den Amazon S3 S3-Speicherort an, an den Sie die vom Quell-Mainframe abgerufenen Datensätze hochgeladen haben.
5. Klicken Sie auf Senden, um den Aufzeichnungsvorgang zu starten.



**Note**

Warten Sie, bis die Aufnahme abgeschlossen ist, bevor Sie die Wiedergabe und den Vergleich durchführen.

## Schritt 12: Wiedergeben und vergleichen

Führen Sie das Testszenario und die Testfälle in der Zielumgebung AWS AWS Blu Age auf Amazon EC2 aus. Beim Anwendungstest werden die in der Wiedergabe erstellten Datensätze erfasst und mit den Referenzdatensätzen verglichen, die auf dem Mainframe aufgezeichnet wurden.

1. Wählen Sie „Wiedergeben und Vergleichen“. Es sollte ungefähr drei Minuten dauern, den CloudFormation Stapel zu erstellen, den Vergleich durchzuführen und den Stapel zu löschen.

Sobald alles abgeschlossen ist, sollten Sie Vergleichsergebnisse mit einigen Unterschieden haben, die bewusst für die Zwecke dieser Demo erstellt wurden.

## AWS Mainframe-Modernisierung, Anwendungstests, unterstützte Datensätze, Codepages

AWS Anwendungstests befinden sich in der Vorschauversion für AWS Mainframe Modernization und können sich ändern. Wir empfehlen, diese Funktion nur mit Testdaten und Anwendungen und nicht in Produktionsumgebungen zu verwenden.

Ermitteln Sie anhand der folgenden Tabelle, ob der Coded Character Set Identifier (CCSID) für Ihre Daten bei AWS Anwendungstests unterstützt wird. Wenn Ihre Daten eine CCSID verwenden, die nicht unterstützt wird, empfehlen wir Ihnen, diese entweder in eine unterstützte CCSID zu konvertieren oder uns um Hilfe zu [bitten](#).

CCSID	Zeichensätze	Beschreibung
37	IBM037, IBM-037, Cp037	Gastgeber: USA, Kanada (ESA), Niederlande, Portugal,

CCSID	Zeichensätze	Beschreibung
		Brasilien, Australien, Neuseeland
273	IBM 273, IBM 273, Cp 273	Gastgeber: Österreich, Deutschland
277	IBM277, IBM-277, Cp277	Gastgeber: Dänemark, Norwegen
278	IBM 278, IBM 278, Cp 278	Gastgeber: Finnland, Schweden
280	IBM280, IBM-280, Cp280	Gastgeber: Italien
284	IBM 284, IBM 284, CP284	Gastgeber: Spanien, Lateinamerika (Spanisch)
285	IBM 285, IBM 285, Cp 285	Gastgeber: Großbritannien
297	IBM 297, IBM 297, Cp 297	Gastgeber: Frankreich
300	IBM-300	JAPAN DB EBCDIC
301	IBM-301	PC-Daten: Japan DB
437	IBM 437, IBM-437, US-ASCII, ASCII, Cp437, US-ASCII	PC-Daten: PC Base USA, viele andere Länder
500	IBM500, IBM-500, Cp500	Gastgeber: Belgien, Kanada (AS/400), Schweiz, International Latin-1
720	IBM-720	MADOS ARABISCH
737	IBM-737, x-IBM 737	MADOS GRIECHISCH
775	IBM775, IBM-775	MADOS BALTISCH

CCSID	Zeichensätze	Beschreibung
808	IBM-808	PC-Daten: Kyrillisch, Russland, mit Euro
813	ISO-8859-7, ISO8859_7	ISO 8859-7: Griechenland
819	ISO-8859-1, ISO8859_1	ISO 8859-1: Latin-1-Länder
833	IBM-833	KOREANISCH EBDDISCH
834	IBM-834, x-IBM 834	KOREANISCHER DB EBCDIC
835	IBM-835	T-CHINESISCHES DB EBCD
836	IBM-836	S-CHINESISCH EBCDISCH
837	IBM-837	S-CHINESISCH EBCDISCH
850	IBM 850, IBM 850, CP850	PC-Daten: Latin-1-Länder
855	IBM 855, IBM 855, Cp 855	PC-Daten: Kyrillisch
856	IBM-856, x-IBM 856, CP856	PC-Daten: Hebräisch
858	IBM 00858, IBM 858, CP 858	PC-Daten: Latin-1-Länder, mit Euro
859	IBM-859	PC-Daten: LATIN-9
860	IBM-860, IBM-860	PC-Daten: Portugiesisch
861	IBM-861, IBM-861	PC-Daten: Island
862	IBM 862, IBM 862, Cp 862	PC-Daten: Hebräisch (Migration)
863	IBM-863, IBM-863	PC-Daten: Kanada
865	IBM 865, IBM 865, Cp 865	PC-Daten: Den/Norwegen

CCSID	Zeichensätze	Beschreibung
866	IBM 866, IBM 866, Cp 866	PC-Daten: Kyrillisch, Russland
867	IBM-867	PC-Daten: Hebräisch mit Euro
870	IBM 870, IBM 870, CP 870	Gastgeber: Latin-2 mehrsprachig
871	IBM 871, IBM-871, Cp 871	Gastgeber: Island
874	x-IBM 874	PC-Daten: Thailändisch
875	IBM-875, x-IBM 875, Cp 875	Gastgeber: Griechenland
897	IBM-897	PC-Daten: Japan SB
912	ISO-8859-2, ISO8859_2	ISO 8859-2: Latin-2 mehrsprachig
915	ISO-8859-5, ISO8859_5	ISO 8859-5: Kyrillisch
916	ISO-8859-8, ISO8859_8	ISO 8859-8: Hebräisch
918	IBM 918, IBM 918, CP 918	Gastgeber: Urdu
920	ISO-8859-9, ISO8859_9	ISO 8859-9: Latin-5 (ECMA-128, Türkei TS-5881)
921	IBM-921, x-IBM 921, CP921	PC-Daten: Lettland, Litauen
922	IBM-922, x-IBM 922, CP922	PC-Daten: Estland
923	ISO-8859-15, Cp923, ISO8859_15_FDIS	ISO 8859-15: Latein-9
924	IBM-924	ISO 8859-15: Latein-9
927	IBM-927	PC-Daten: T-Chinesisch

CCSID	Zeichensätze	Beschreibung
930	IBM-930, x-IBM930, CP930	Katakana-Host: erweitertes SBCS. Kanji-Host: DBCS mit 4370 benutzerdefinierten Zeichen
932	IBM-932	PC-Daten: Japan Mix
933	IBM-933, x-IBM 933, CP933	Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerdefinierten Zeichen und 11172 vollständigen Hangul-Zeichen
935	IBM-935, x-IBM 935, CP935	Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerdefinierten Zeichen.
937	IBM-937, x-IBM 937, CP937	Host: Erweitertes SBCS. Host: DBCS mit 6204 benutzerdefinierten Zeichen
939	IBM-939, x-IBM 939, CP939	Latin Host: erweitertes SBCS. Kanji-Host: DBCS mit 4370 benutzerdefinierten Zeichen.
942	IBM-942, IBM-942C, X-IBM 942, X-IBM 942C, Cp942, Cp942C	PC-Daten: Erweitertes SBCS. PC-Daten: DBCS mit 1880 benutzerdefinierten Zeichen
943	IBM-943, IBM-943C, Shift_JIS , Fenster-31j, Fenster-932, X-IBM943, X-IBM943c, Cp943, Cp943C, MS932	PC-Daten: SBCS. PC-Daten: DBCS für eine offene Umgebung, einschließlich 1880 benutzerdefinierter IBM-Zeichen
947	IBM-947	T-CHINESE BIG-5

CCSID	Zeichensätze	Beschreibung
948	IBM-948, x-IBM 948, CP948	PC-Daten: Erweitertes SBCS. PC-Daten: DBCS mit 6204 benutzerdefinierten Zeichen
949	IBM-949, IBM-949C, X-IBM 949, X-IBM 949C, CP949, Cp949C	IBM KS Code - PC-Daten: SBCS. IBM KS-Code — PC-Daten: DBCS mit 1880 benutzerdefinierten Zeichen
950	Big5, IBM-950, x-IBM 950, Cp950	PC-Daten: SBCS (IBM BIG5). PC-Daten: DBCS einschließlich 13493 CNS, 566 von IBM ausgewählt, 6204 benutzerdefinierte Zeichen
951	IBM-951	PC-Daten: IBM KS
954	EUC-JP, IBM-954, IBM-954C	G0: JIS X201 Römisch. G1: JIS X208-1990. G1: JIS X201 Katakana. G1: JIS X212
964	EUC-TW, IBM-964, x-IBM964, CP964	G0: ASCII. G1: CNS 11643 Ebene 1. G1: CNS 11643 Flugzeug 2.
970	EUC-KR, X-IBM 970, CP970	G0: ASCII. G1: KSC X5601-1989 einschließlich 1880 benutzerdefinierter Zeichen
971	IBM-971	KOREANISCHE EUC
1006	IBM-1006, x-IBM 1006, CP1006	ISO-8: Urdu
1025	IBM-1025, x-IBM1025, CP1025	Gastgeber: Kyrillisch, mehrsprachig

CCSID	Zeichensätze	Beschreibung
1026	IBM 1026, IBM 1026, Cp 1026	Gastgeber: Latin-5 (Türkei)
1027	IBM-1027	JAPANISCHES LATEINISC HES EBCD
1041	IBM-1041	PC-Daten: Japan
1043	IBM-1043	PC-Daten: T-Chinesisch
1046	IBM-1046, IBM-1046S, x- IBM1046	ARABISCH - PC
1047	IBM1047, IBM-1047	Gastgeber: Latin-1
1051	HP-Roman 8	HP-EMULATION
1088	IBM-1088	PC-Daten: Korea KS
1089	ISO-8859-6, ISO8859_6	ISO 8859-6: Arabisch
1097	IBM-1097, x-IBM 1097, CP1097	Gastgeber: Farsi
1098	IBM-1098, x-IBM 1098, CP1098	PC-Daten: Farsi
1112	IBM-1112, x-IBM1112, Cp1112	Gastgeber: Lettland, Litauen
1114	IBM-1114	PC-Daten: T-CH SB
1115	IBM-1115	PC-Daten: S-CH GB
1122	IBM-1122, x-IBM1122, Cp1122	Gastgeber: Estland
1123	IBM-1123, x-IBM1123, Cp1123	Gastgeber: Kyrillische Ukraine

CCSID	Zeichensätze	Beschreibung
1124	IBM-1124, x-IBM 1124, Cp1124	8-Bit: Kyrillisch, Weissrussland
1140	IBM 01140, IBM 1140, Cp 1140	Gastgeber: USA, Kanada (ESA), Niederlande, Portugal, Brasilien, Australien, Neuseeland, mit Euro
1141	IBM01141, IBM-1141, Cp1141	Gastgeber: Österreich, Deutschland, mit Euro
1142	IBM 01142, IBM 1142, Cp 1142	Gastgeber: Dänemark, Norwegen, mit Euro
1143	IBM 01143, IBM 1143, Cp 1143	Gastgeber: Finnland, Schweden, mit Euro
1144	IBM01144, IBM-1144, Cp1144	Gastgeber: Italien, mit Euro
1145	IBM 01145, IBM 1145, Cp 1145	Gastgeber: Spanien, Lateinamerika (Spanisch), mit Euro
1146	IBM 01146, IBM 1146, Cp 1146	Gastgeber: Großbritannien, mit Euro
1147	IBM 01147, IBM 1147, Cp 1147	Gastgeber: Frankreich, mit Euro
1148	IBM 01148, IBM 1148, Cp 1148	Gastgeber: Belgien, Kanada (AS/400), Schweiz, International Latin-1, mit Euro
1149	IBM 01149, IBM 1149, Cp 1149	Gastgeber: Island, mit Euro



CCSID	Zeichensätze	Beschreibung
1200	UTF-16BE	Unicode mit dem Zeichensatz 65535. In Ermangelung einer Byte-Order-Marke (BOM) wird davon ausgegangen, dass es sich um UTF-16 BE (Big-Endian) handelt.
1202	UTF-16LE	UTF-16 LE mit IBM PUA
1204	UTF-16	UTF-16 mit IBM PUA
1208	UTF-8, UTF-8J, UTF-8	Unicode mit dem Zeichensatz 65535. UTF-8.
1232	UTF-32BE	UTF-32 BE mit IBM PUA
1234	UTF-32LE	UTF-32 LE mit IBM PUA
1236	UTF-32	UTF-32 mit IBM PUA
1351	IBM-1351	JAPAN GEÖFFNET
1362	IBM-1362	KOREANISCH MS-WIN
1363	IBM-1363, IBM-1363C, Windows-949, MS949	PC-Daten: MS Windows Korean SBCS. PC-Daten: MS Windows Korean DBCS, einschließlich 11172 vollständiger Hangeul-Version
1364	IBM-1364	Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerdefinierten Zeichen und 11172 vollständigen Hangeul-Zeichen

CCSID	Zeichensätze	Beschreibung
1370	IBM-1370	PC-Daten: Erweitertes SBCS, mit Euro. PC-Daten: DBCS mit 6204 benutzerdefinierten Zeichen, mit Euro
1371	IBM-1371	Gastgeber: Extended SBCS, mit Euro. Host: DBCS mit 6204 benutzerdefinierten Zeichen, mit Euro
1375	Big5-HKSCS	Gemischter Big-5 Ext für HKSCS
1380	IBM-1380	PC-Daten: S-CH GB
1381	IBM-1381, x-IBM 1381, CP1381	PC-Daten: Erweitertes SBCS (IBM GB). PC-Daten: DBCS (IBM GB), einschließlich 31 von IBM ausgewählter, 1880 benutzerdefinierter Zeichen
1382	IBM-1382	S-CHINESISCHE EUC
1383	EUC-CN, GB2312, IBM-1383, x-IBM1383, Cp1383	G0: ASCII. G1: Satz GB 2312-80
1385	IBM-1385	PC-Daten: S-CH GBK
1386	GBK, IBM-1386, Windows-936, MS936	PC-Daten: S-chinesisches GBK und T-chinesisches IBM BIG-5. PC-Daten: S-chinesisches GBK
1388	IBM-1388	Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerdefinierten Zeichen

CCSID	Zeichensätze	Beschreibung
1390	IBM-1390	Katakana Host: erweitertes SBCS, mit Euro. Kanji-Host: DBCS mit 6205 benutzerdefinierten Zeichen
1399	IBM-1399	Latin Host: erweitertes SBCS, mit Euro. Kanji Host: DBCS mit 4370 benutzerdefinierten Zeichen, mit Euro
5050	JIS0201, JIS0208, JIS0212, JIS0201, JIS0208, JIS0212	G0: JIS X201 römisch. G1: JIS X208-1990. G1: JIS X201 Katakana. G1: JIS X212
5054	ISO-2022-JP	JAPANISCHES TCP
5346	Fenster-1250, Cp1250	MS Windows: Latin-2, Version 2 mit Euro
5347	Fenster-1251, Cp1251	MS Windows: Kyrillisch, Version 2 mit Euro
5348	Fenster-1252, Cp1252	MS Windows: Latin-1-Länder, Version 2 mit Euro
5349	Fenster-1253, Cp1253	MS Windows: Griechenland, Version 2 mit Euro
5350	Fenster-1254, Cp1254	MS Windows: Türkei, Version 2 mit Euro
5351	Fenster-1255, Cp1255	MS Windows: Hebräisch, Version 2 mit Euro
5352	Fenster-1256, Fenster-1256S, Cp1256	MS Windows: Arabisch, Version 2 mit Euro

CCSID	Zeichensätze	Beschreibung
5353	Fenster-1257, Cp1257	MS Windows: Baltic Rim, Version 2 mit Euro
5354	Fenster-1258, Cp1258	MS Windows: Vietnamesisch, Version 2 mit Euro
5488	GB18030	GB18030, 1-Byte-Daten GB18030, 2-Byte-Daten GB18030, 4-Byte-Daten
9030	IBM-838, Cp 838	Gastgeber: Extended SBCS in Thailand
9066	IBM-874, Cp 874	PC-Daten: Erweitertes thailändisches SBCS
9400	CESU-8	CESU-8 mit IBM PUA
25546	ISO-2022-KR	KOREANISCHER TCP
33722	IBM-33722, IBM-33722C	IBM EUCJP

# Dateiübertragung bei der AWS Mainframe-Modernisierung

Mit AWS Mainframe Modernization File Transfer können Sie Mainframe-Datensätze für Anwendungsfälle zur Mainframe-Modernisierung, -Migration und Erweiterung nach Amazon S3 übertragen und konvertieren.

## Themen

- [Was ist AWS Mainframe Modernization File Transfer?](#)
- [Installieren Sie einen File Transfer Agent](#)
- [Endpunkte für die Datenübertragung](#)
- [Aufgaben übertragen](#)
- [Tutorial: Erste Schritte mit AWS Mainframe Modernization File Transfer](#)
- [Von AWS Mainframe Modernization File Transfer unterstützte Quell- und Zielkodierungen](#)

## Was ist AWS Mainframe Modernization File Transfer?

Mit AWS Mainframe Modernization File Transfer können Sie Datensätze und Dateien mit einem vollständig verwalteten Service übertragen und konvertieren, um Anwendungsfälle für Modernisierung, Migration und Erweiterung für den AWS Mainframe Modernization Service und Amazon S3 zu beschleunigen und zu vereinfachen.

## Themen

- [Vorteile von AWS Mainframe Modernization File Transfer](#)
- [So funktioniert AWS Mainframe Modernization File Transfer](#)

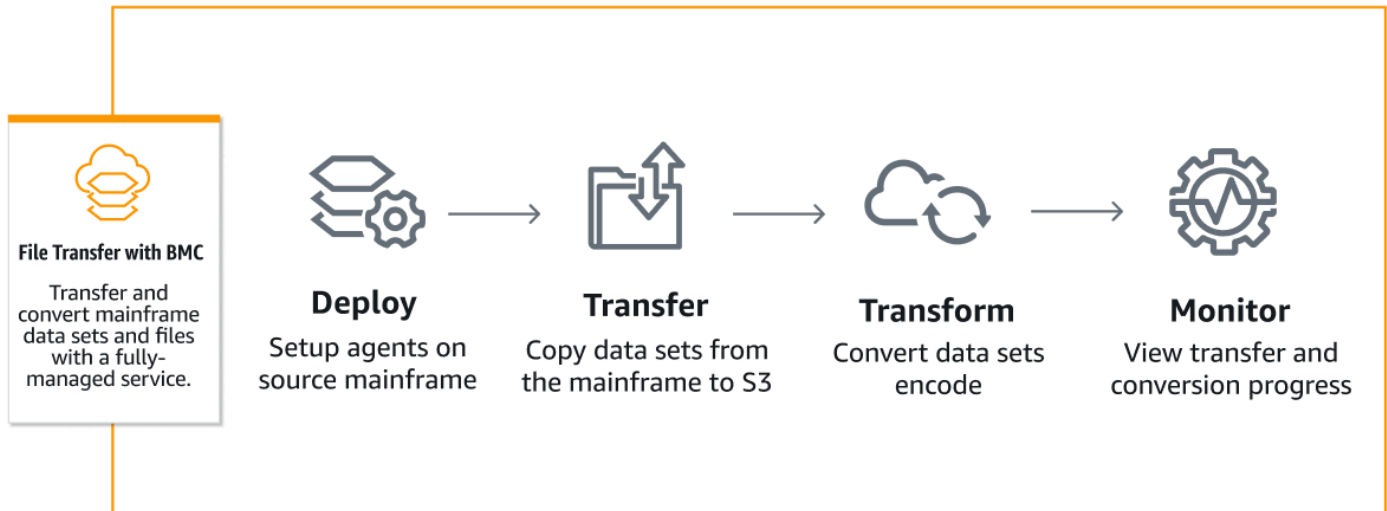
## Vorteile von AWS Mainframe Modernization File Transfer

Mit AWS Mainframe Modernization File Transfer können Sie Datensätze vom Mainframe nach Amazon S3 übertragen. Zu den Vorteilen gehören:

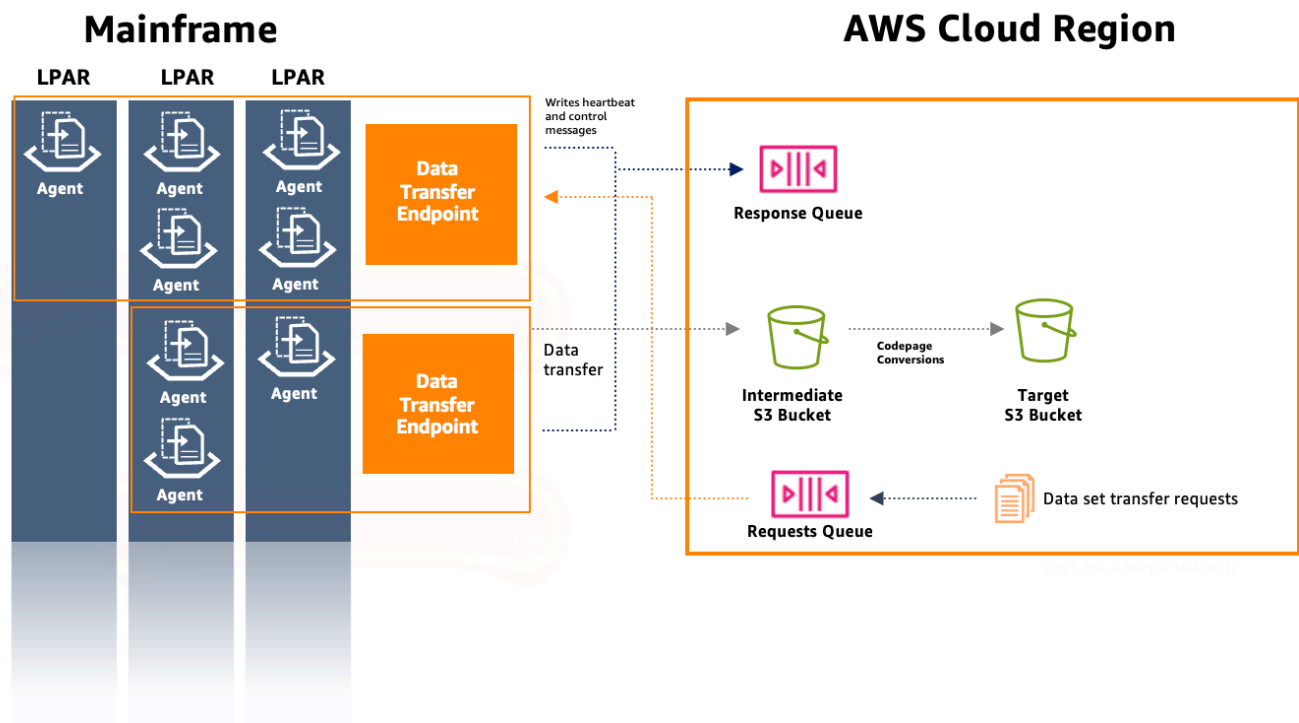
- Entdeckung von Mainframe-Quelldatensätzen und Artefakten
- Automatisierte Übertragungen und Konvertierung von Datensätzen
- Skalierbarkeit, Effizienz und Geschwindigkeit für schnellere Datenübertragungen an AWS

# So funktioniert AWS Mainframe Modernization File Transfer

Die folgende Abbildung gibt einen Überblick darüber, wie AWS Mainframe Modernization File Transfer auf konzeptioneller Ebene funktioniert.



Die folgende Abbildung gibt einen Überblick über die Architektur der Dateiübertragungsfunktion von AWS Mainframe Modernization.



# Installieren Sie einen File Transfer Agent

Folgen Sie dieser Anleitung step-by-step , um alle Voraussetzungen für die Installation eines Agenten auf dem Quell-Mainframe zu erfüllen und den Agenten zu konfigurieren.

## Themen

- [Schritt 1: Melden Sie sich bei ISPF an](#)
- [Schritt 2: Ordnen Sie dem z/FS einen Datensatz zu](#)
- [Schritt 3: Formatieren Sie den Datensatz als z/FS](#)
- [Schritt 4: Definieren Sie das Dateisystem auf z/OS](#)
- [Schritt 5: Mounten Sie das Dateisystem](#)
- [Schritt 6: Überprüfen Sie den Mount](#)
- [Schritt 7: Geben Sie OMVs ein](#)
- [Schritt 8: Legen Sie die Umgebungsvariable für das Installationsverzeichnis des Agenten fest](#)
- [Schritt 9: Legen Sie die Umgebungsvariable für das Arbeitsverzeichnis fest](#)
- [Schritt 10: Erstellen Sie das Arbeitsverzeichnis](#)
- [Schritt 11: Kopieren Sie das Tar-Paket AWS Mainframe Modernization in das Arbeitsverzeichnis auf z/OS](#)
- [Schritt 12: Gehen Sie vom Root-Benutzer aus](#)
- [Konfigurieren Sie Berechtigungen und STC](#)
- [Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten](#)
- [Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll](#)
- [Konfiguration des Agenten](#)

## Schritt 1: Melden Sie sich bei ISPF an

Loggen Sie sich in Ihre ISPF-Sitzung (Interactive System Productivity Facility) ein. Dies erfolgt normalerweise über einen 3270-Terminalemulator.

## Schritt 2: Ordnen Sie dem z/FS einen Datensatz zu

Verwenden Sie das Dataset-Hilfsprogramm von ISPF und weisen Sie dem z/FS einen neuen Datensatz zu. In der Regel erfolgt dies im Programm zur Datensatzliste in Schritt 1.

1. Gehen Sie zu Option 3.4 (Dienstprogramme -> Datensatz).
2. Drücken Sie die Taste 'C', um einen neuen Datensatz zu erstellen.
3. Geben Sie den Namen des Datensatzes ein (zum Beispiel „YourHLQ.M2Agent.ZFS“).
4. Geben Sie als Datensatztyp „Großformat“ mit einer Primärgröße von 1000 Zylindern und einer Sekundärgröße von 200 Zylindern an.
5. Stellen Sie die Datensatzorganisation (DSORG) auf PS und das Datensatzformat (RECFM) auf „U“ (Undefiniert) ein.
6. Schließen Sie den Erstellungsprozess ab.

### Schritt 3: Formatieren Sie den Datensatz als z/FS

Nachdem Sie den Datensatz erstellt haben, formatieren Sie ihn als z/FS-Dateisystem.

Eine Möglichkeit, dies zu tun, ist die Verwendung der folgenden Job Control Language (JCL):

```
//FORMAT EXEC PGM=IOEAGFMT,PARM='AGGRNAME(yourhlq.M2AGENT.ZFS),FORMAT,AGGRSIZE(1200)'  
//SYSPRINT DD SYSOUT=A
```

Reichen Sie diesen Job ein und überprüfen Sie, ob er erfolgreich abgeschlossen wurde.

### Schritt 4: Definieren Sie das Dateisystem auf z/OS

Definieren Sie z/FS für z/OS mithilfe eines DEF FILESYSTEM Befehls oder über einen Batch-Job.

Verwenden Sie den folgenden Befehl:

```
DEF FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(R/W) MOUNTPPOINT('/usr/lpp/aws/m2-agent')
```

#### Note

Für diesen Schritt sind Berechtigungen auf Systemebene erforderlich.

### Schritt 5: Mounten Sie das Dateisystem

Um das Dateisystem zu mounten, verwenden Sie den MOUNT Befehl. Sie können das Dateisystem in der Befehlszeile in ISPF oder im Batch mounten.



**Beispielsweise:**

```
MOUNT FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/usr/lpp/aws/m2-agent')
```

## Schritt 6: Überprüfen Sie den Mount

Stellen Sie sicher, dass das Dateisystem korrekt gemountet ist. Verwenden Sie dazu einen `D OMVS, A` Befehl oder überprüfen Sie es im Unix System Service (USS).

## Schritt 7: Geben Sie OMVs ein

Verwenden Sie den folgenden Befehl, um OMVs einzugeben:

```
TSO OMVS
```

## Schritt 8: Legen Sie die Umgebungsvariable für das Installationsverzeichnis des Agenten fest

Verwenden Sie den folgenden Befehl, um die Umgebung für das Installationsverzeichnis des Agenten festzulegen:

```
export AGENT_DIR=/usr/lpp/aws/m2-agent
```

## Schritt 9: Legen Sie die Umgebungsvariable für das Arbeitsverzeichnis fest

Verwenden Sie den folgenden Befehl, um die Umgebungsvariable für das Arbeitsverzeichnis festzulegen:

```
export WORK_DIR=$AGENT_DIR/tmp
```

## Schritt 10: Erstellen Sie das Arbeitsverzeichnis

Verwenden Sie den folgenden Befehl, um die Arbeitsverzeichnisumgebung festzulegen:

```
mkdir -p $WORK_DIR
```

## Schritt 11: Kopieren Sie das Tar-Paket AWS Mainframe Modernization in das Arbeitsverzeichnis auf z/OS

Wenn Sie FTP oder eine andere Übertragungsmethode verwenden, stellen Sie sicher, dass die TAR-Datei im Binärmodus übertragen wird.

## Schritt 12: Gehen Sie vom Root-Benutzer aus

Verwenden Sie den folgenden Befehl, um einen Root-Benutzer anzunehmen:

```
su
```

Gehen Sie wie folgt vor, um die Agenteninstallation abzuschließen:

### Note

Sie müssen den Root-Benutzer annehmen, bevor Sie mit diesen Schritten fortfahren können.

1. Setzen Sie die Umgebungsvariable `m2-agent-Version` mit dem folgenden Befehl auf die Version, die gerade installiert wird:

```
export M2_AGENT_VERSION=1.0.0
```

2. Extrahieren Sie das Agenten-Tar-Paket mit dem folgenden Befehl:

```
tar -xpf m2-agent-package-$M2_AGENT_VERSION.tar -C $AGENT_DIR
```

3. Erstellen Sie mit dem folgenden Befehl einen `current-version` symbolischen Link zum aktuellen Agenten-Installationsverzeichnis:

```
ln -s $AGENT_DIR/m2-agent-v$M2_AGENT_VERSION $AGENT_DIR/current-version
```

4. Aktualisieren und abschicken `CPY#PDS`, um die File Transfer Agent-Datensätze zu erstellen.

### Note

JCL verwendet den HLQ `SYS2.AWS.M2`.

Um den File Transfer Agent zu erstellen, legen Sie die Parameterzeilen 000006-000012 fest. Aktualisieren Sie außerdem die drei symbolischen Variablen HLQ, und VOLSER, die später in der AGNTPATH JCL verwendet werden sollen:

```
oedit $AGENT_DIR/current-version/installation/CPY#PDS
submit $AGENT_DIR/current-version/installation/CPY#PDS
```

### Note

Diese JCL ist auf die Einrichtung bestimmter Aspekte der Agenteninstallation auf dem Mainframe zugeschnitten. Sie weist die erforderlichen Datensätze zu und kopiert dann bestimmte Dateien aus dem Unix-Dateisystem in diese Datensätze.

## Konfigurieren Sie Berechtigungen und STC

1. Aktualisieren Sie eines der `SYS2.AWS.M2.SAMPLIB(SEC#RACF)` (zum Einrichten von RACF-Berechtigungen) oder `SYS2.AWS.M2.SAMPLIB(SEC#TSS)` (zum Einrichten von TSS-Berechtigungen) und reichen Sie es gemäß den jeweiligen Anweisungen ein. Diese Mitglieder wurden im vorherigen CPY#PDS Schritt erstellt.
2. Aktualisieren Sie den PWD-Export in der `SYS2.AWS.M2.SAMPLIB(M2AGENT) STC JCL`, falls der Standard-Verzeichnispfad (`/usr/lpp/aws/m2-agent`) für den File Transfer Agent geändert wurde.
3. Aktualisieren und kopieren Sie die `SYS2.AWS.M2.SAMPLIB(M2AGENT) JCL` nach `SYS1.PROCLIB`
4. Fügen Sie `SYS2.AWS.M2.LOADLIB` es mit dem folgenden Befehl zur APF-Liste hinzu:

```
SETPROG APF ADD DSNAME(SYS2.AWS.M2.LOADLIB) SMS
```

5. Legen Sie die Gruppe und den Besitzer des Agenten `logs` und `diag` der Ordner auf den Benutzer/die Gruppe des Agenten fest (`M2USER/M2GROUP`). Verwenden Sie den folgenden Befehl:

```
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/logs
```

```
chown -R M2USER:M2GROUP $AGENT_DIR/current-version/diag
```

## Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten

Sie müssen den Mainframe-Agenten erstellen, den ein IAM-Benutzer benötigt, um Antwort- und Anforderungswarteschlangen zu verwenden und Datensätze in Amazon S3 S3-Buckets zu speichern.

Gehen Sie beim Erstellen dieses Benutzers wie folgt vor:

1. Wählen Sie in den Berechtigungsoptionen die Option Richtlinien direkt anhängen aus.
2. Sobald der Benutzer erstellt wurde, öffnen Sie den Tab Sicherheitsanmeldedaten und erstellen Sie einen Zugriffsschlüssel. Weitere Informationen zum Erstellen eines IAM-Zugriffsschlüssels finden Sie unter [Zugriffsschlüssel für IAM-Benutzer verwalten](#).
3. Wählen Sie im Abschnitt Zugriffsschlüssel die Option Andere aus, wenn Sie nach einem Anwendungsfall gefragt werden.

### Note

Speichern Sie den Zugriffsschlüssel und den geheimen Zugriffsschlüssel, die auf der letzten Seite des Assistenten zur Erstellung des Zugriffsschlüssels angezeigt werden, bevor Sie Fertig wählen. Diese Schlüssel werden zur Konfiguration des Mainframe-Agenten verwendet.

### Note

Speichern Sie den IAM-Benutzer-ARN, der zum Einrichten einer Vertrauensbeziehung mit einer IAM-Rolle verwendet wurde.

## Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll

Sie erstellen eine neue IAM-Rolle mit einer benutzerdefinierten Vertrauensrichtlinie für den Entitätstyp Vertrauenswürdig. Die Richtlinie verwendet die folgende Vorlage:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "sts:AssumeRole",  
      "Resource": "*" }  
    ]  
}
```

```
{
  "Sid": "DataTransferEndpointAgentSqsReceive",
  "Effect": "Allow",
  "Action": [
    "sqs:DeleteMessage",
    "sqs:ReceiveMessage"
  ],
  "Resource": "<data-transfer-endpoint-request-queue-arn>"
},
{
  "Sid": "DataTransferEndpointS3",
  "Effect": "Allow",
  "Action": "s3:PutObject",
  "Resource": "<data-transfer-endpoint-intermediate-bucket-arn>/*"
},
{
  "Sid": "DataTransferEndpointAgentSqsSend",
  "Effect": "Allow",
  "Action": "sqs:SendMessage",
  "Resource": "<data-transfer-endpoint-response-queue-arn>"
},
{
  "Sid": "DataTransferEndpointAgentKmsDecrypt",
  "Effect": "Allow",
  "Action": "kms:Decrypt",
  "Resource": "<kms-key-id>"
}
]
```

Wobei gilt:

- `request-queue-arn` und `response-queue-arn` sind der ARN der Amazon SQS SQS-Anforderungswarteschlange, die während der Initialisierung des Datenübertragungsendpunkts erstellt wurde.
- `transfer-bucket-arn` ist der ARN des zuvor erstellten Transfer-Buckets.

#### Note

Sie können all diese Werte mit der AWS-Konsole nachschlagen.

**Note**

Speichern Sie den Rollennamen, den Sie später zur Konfiguration des Mainframe-Agenten verwenden werden.

## Konfiguration des Agenten

So konfigurieren Sie den File Transfer Agent:

1. Navigieren Sie zu `$AGENT_DIR/current-version/config`.
2. Bearbeiten Sie die Konfigurationsdatei des Agenten `application.properties`, um mit dem folgenden Befehl eine Umgebungskonfiguration hinzuzufügen:

```
oedit $AGENT_DIR/current-version/config/application.properties
```


Beispielsweise:

```
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>
```

Wobei gilt:

- `AWS_ACCOUNT_ID` ist die ID des Kundenkontos.
- `AWS_IAM_ROLE_NAME` ist der Name der IAM-Rolle, die in der [the section called “Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll”](#) erstellt wurde.
- `AWS_IAM_ROLE_ACCESS_KEY` ist der Zugriffsschlüssel des IAM-Benutzers, der in erstellt wurde. [the section called “Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten”](#)
- `AWS_IAM_ROLE_SECRET_KEY` ist der geheime Zugriffsschlüssel für den IAM-Benutzer, der in erstellt wurde. [the section called “Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten”](#)

- `AWS_S3_BUCKET_NAME` ist der Name des Transfer-Buckets, der mit dem Datenübertragungsendpunkt erstellt wurde.
- `AWS_REGION` ist die Region, in der Sie den File Transfer Agent konfigurieren.

 Note

Es kann mehrere solcher Abschnitte geben, sofern der Index in Klammern — `[0]` — für jeden Abschnitt inkrementiert wird.

Sie müssen den Agenten neu starten, damit die Änderungen wirksam werden.

### Voraussetzungen

1. Wenn ein Parameter hinzugefügt oder entfernt wird, muss der Agent gestoppt und gestartet werden. Starten Sie den File Transfer Agent mit dem folgenden Befehl in der CLI:

```
/S M2AGENT
```

Verwenden Sie den folgenden Befehl in der CLI, um den M2-Agenten zu beenden:

```
/P M2AGENT
```

2. Sie können den File Transfer Agent in mehrere Regionen und Konten übertragen lassen, AWS indem Sie mehrere Umgebungen definieren.

```
#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
```

## Endpunkte für die Datenübertragung

Datenübertragungsendpunkte ermöglichen die Konnektivität mit dem Quell-Mainframe und unterstützen hohe Verfügbarkeit, Skalierbarkeit und eine optimierte Verwaltung von Agenten. Einzelne Agenten werden auf Mainframe-LPARs installiert und können zu einem Datenübertragungsendpunkt zusammengefasst werden. Wenn eine Anfrage zur Übertragung eines Datensatzes gestellt wird, kümmert sich ein Agent am Datenübertragungsendpunkt um diese spezifische Übertragung. Um Datenübertragungen zu initiieren, muss mindestens ein Agent am Datenübertragungsendpunkt online sein.

Bei diesem Verfahren wird davon ausgegangen, dass Sie die Schritte [unter File Transfer Agent konfigurieren auf dem Quell-Mainframe](#) abgeschlossen haben. [Einrichtung der AWS Mainframe-Modernisierung](#)

## Erstellen Sie einen Datenübertragungsendpunkt


Gehen Sie in der AWS Mainframe-Modernisierungskonsole wie folgt vor, um Datenübertragungsendpunkte für die Dateiübertragung zu erstellen.

Um einen Datenübertragungsendpunkt zu erstellen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie Dateien von Ihrem Mainframe in einen Amazon S3 S3-Bucket übertragen möchten.
3. Wählen Sie auf der Seite Datenübertragungsendpunkte unter Dateiübertragung die Option Datenübertragungsendpunkt erstellen aus.
4. Lesen Sie auf der Seite mit den Voraussetzungen für Datenübertragungsendpunkte alle Anweisungen, um sicherzustellen, dass Sie diese Schritte auf dem Quell-Mainframe abgeschlossen haben. Wählen Sie nach der Bestätigung Weiter aus.
5. Fügen Sie auf der Seite Datenübertragungsendpunkt konfigurieren grundlegende Informationen für Ihren Datenübertragungsendpunkt hinzu.




1. Geben Sie im Abschnitt mit den grundlegenden Informationen den Namen Ihres Datenübertragungsendpunkts ein.

 Note

Der Name des Datenübertragungsendpunkts muss mit dem Sysplex-Namen übereinstimmen, es sei denn, Sie geben in der Agentenkonfiguration einen komplexen Namen an.


2. Eine optionale Beschreibung.
3. Der KMS-Schlüssel, der zum Verschlüsseln des Geheimnisses verwendet wurde.

 Note

Sie müssen die folgende ressourcenbasierte Richtlinie für KMS hinzufügen, damit der AWS Mainframe-Modernisierungsdienst diese Schlüssel lesen und für die Verschlüsselung/Entschlüsselung verwenden kann:

```
{
  "Sid" : "Enable AWS M2 Permissions",
  "Effect" : "Allow",
  "Principal" : {
    "Service" : "m2.amazonaws.com"
  },
  "Action" : [
    "kms:Encrypt",
    "kms:Decrypt"
  ],
  "Resource" : "*"
}
```

4. Geben Sie den S3-Speicherort für Zwischendaten an. Dabei handelt es sich um den S3-Zwischenstandort, an dem übertragene Datensätze vom Mainframe gespeichert werden, bevor sie konvertiert und in den Amazon S3 S3-Ziel-Bucket übertragen werden.

 Note

Es wird empfohlen, einen neuen Amazon S3 S3-Bucket für Ihre Übertragungsaufgaben zu erstellen. Weitere Informationen finden Sie unter [Bucket erstellen](#). Sie können auch Ihre vorhandenen Amazon S3 S3-Buckets durchsuchen, indem Sie die Option S3 durchsuchen wählen.

5. Nachdem Sie die erforderlichen Felder eingegeben haben, wählen Sie Weiter.
6. Prüfen Sie auf der Seite Datenübertragungsendpunkt überprüfen und erstellen, ob Sie die Voraussetzungen erfüllt haben, und überprüfen Sie die grundlegenden Informationen. Wählen Sie nach der Bestätigung die Option Datenübertragungsendpunkt erstellen aus.

Sie werden zur Übersichtsseite der Datenübertragungsendpunkte weitergeleitet, auf der Sie die Liste aller Datenübertragungsendpunkte sehen können. Sie können auch die Datenübertragungsendpunkte sehen, die verfügbar sind oder ausgefallen sind.

Sie können Datenübertragungsendpunkte auch nach Namen suchen und auf zusätzliche Informationen für jeden verfügbaren Agenten zugreifen.

## Aufgaben übertragen

Übertragungsaufgaben werden verwendet, um die Datensätze zu spezifizieren, die vom Mainframe zu Amazon S3 übertragen werden sollen, und ermöglichen es Ihnen, die Optionen für die Codepage-Konvertierung auszuwählen.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter abgeschlossen [Einrichtung der AWS Mainframe-Modernisierung](#) und erstellt [the section called "Endpunkte für die Datenübertragung"](#) haben.

### Themen

- [Übertragungsaufgaben erstellen](#)
- [Übertragungsaufgaben anzeigen](#)

## Übertragungsaufgaben erstellen

Gehen Sie in der AWS Mainframe-Modernisierungskonsole wie folgt vor, um Übertragungsaufgaben in File Transfer zu erstellen.

Um eine Übertragungsaufgabe zu erstellen

### Important


Sie benötigen mindestens einen Datenübertragungsendpunkt, um neue Übertragungsaufgaben erstellen zu können.

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie Dateien von Ihrem Mainframe in einen Amazon S3 S3-Bucket übertragen möchten.
3. Auf der Seite Übertragungsaufgaben können Sie einen beliebigen Datenübertragungsendpunkt auswählen, um Übertragungsaufgaben zu erstellen.
4. Richten Sie auf der Seite Übertragungsaufgabe erstellen die Eigenschaften für Ihre Übertragungsaufgabe ein. Wenn Sie noch keine Übertragungsaufgaben erstellt haben, können Sie Ihre erste Aufgabe erstellen, indem Sie die Option Übertragungsaufgabe erstellen wählen.
  - Geben Sie auf dieser Seite die grundlegenden Informationen Ihrer Übertragungsaufgabe ein, einschließlich des Namens der Übertragungsaufgabe, der Beschreibung und des geheimen Schlüssels.

### Note

- Verschlüsseln Sie das Geheimnis mit dem KMS-Schlüssel, der für den Datenübertragungsendpunkt definiert ist. Das Geheimnis sollte die Mainframe-Anmeldeinformationen enthalten, die für den Zugriff auf Datensätze auf dem Mainframe mithilfe der `userId` Schlüssel und erforderlich sind. `password` Weitere Informationen finden Sie im [geheimen AWS Secrets Manager](#).
- Sie müssen den geheimen Schlüssel mit der folgenden ressourcenbasierten Richtlinie konfigurieren, damit der AWS Mainframe Modernization Service darauf zugreifen kann, um Datenübertragungsaufgaben auszuführen.

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

 Note

Die derzeit maximal unterstützte Datensatzgröße für die Übertragung beträgt 90 GB.

- Wählen Sie als Nächstes den Zielort des Amazon S3 S3-Buckets aus, an den die Zieldatensätze vom Mainframe übertragen werden sollen.
  - Der zuvor gewählte Datenübertragungsendpunkt wird ausgewählt. Sie können auch einen anderen Endpunkt aus den verfügbaren Endpunkten auswählen.
5. Wählen Sie Weiter aus.
  6. Geben Sie auf der Seite Datensätze hinzufügen Ihre Abfrage in das Feld Mainframe nach Datensätzen suchen ein, um den Mainframe nach Datensätzen zu durchsuchen, die in Ihre Übertragungsaufgabe aufgenommen werden sollen. Wählen Sie Datensätze anzeigen aus.

Die folgenden Platzhaltersymbole können als Teil der Suchkriterien für Datensätze für Mainframe verwendet werden:

- Ein einzelnes Sternchen (\*) als Qualifikator (zwischen Perioden oder nach dem letzten Zeitraum) entspricht einem einzelnen Qualifikator an dieser Position.
- Ein einzelnes Sternchen (\*) innerhalb eines Qualifikators entspricht null oder mehr Zeichen an dieser Position.
- Ein doppeltes Sternchen (\*\*) als Qualifikator (zwischen Perioden oder nach der letzten Periode) entspricht null oder mehr Qualifizierern an dieser Position.
- Ein doppeltes Sternchen (\*\*) innerhalb eines Qualifikators ist keine gültige Abfrage.

- Ein einzelnes Prozentzeichen (%) entspricht einem beliebigen alphanumerischen oder nationalen Zeichen an dieser Position. Sie können in jedem Qualifikator bis zu acht Prozentzeichen verwenden.

#### Important

Wir empfehlen, Ihre Suchkriterien immer mit einem Punkt gefolgt von einem doppelten Sternchen (\*\*) zu beenden und die Suche dann bei Bedarf weiter zu verfeinern.

Weitere Informationen zu Platzhalterregeln finden Sie in der IBM-Dokumentation unter [Filtern von Datensatznamen](#).

7. Diese Datensätze werden in den Bereich Mainframe-Datensätze geladen, wo Sie nach einem oder mehreren Datensätzen suchen oder diese auswählen können, für die Sie Codepage-Konvertierungen konfigurieren möchten. Diese ausgewählten Datensätze werden im Abschnitt Hinzugefügte Datensätze angezeigt.

#### Note

Sie können Datensätze aus mehreren Suchanfragen auswählen und sie zu Ihrer Übertragungsaufgabe hinzufügen.

8. Im Abschnitt Hinzugefügte Datensätze müssen Sie die Quellcodepage und die Zielcodepage für jeden von Ihnen ausgewählten Datensatz manuell auswählen. Die Quellcodepage ist das Quelldatensatzformat, und die Zielcodepage ist das Zieldatensatzformat, das verwendet wird, um die Datensätze zu konvertieren und im Amazon S3 S3-Ziel-Bucket zu speichern.
9. Nachdem Sie die Quell- und Zielcodepages bestätigt haben, wählen Sie Weiter.
10. Auf der Seite Überprüfen und erstellen können Sie die Informationen für Ihre Übertragungsaufgabe überprüfen oder bearbeiten.
11. Wählen Sie dann Übertragungsaufgabe erstellen aus.

## Übertragungsaufgaben anzeigen

Um Übertragungsaufgaben in File Transfer anzuzeigen, müssen Sie diese Schritte in der AWS Mainframe-Modernisierungskonsole ausführen.

## Um Übertragungsaufgaben anzuzeigen

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie Dateien von Ihrem Mainframe in einen Amazon S3 S3-Bucket übertragen möchten.
3. Wählen Sie auf der Seite Übertragungsaufgaben den Datenübertragungsendpunkt aus, um Ihre Übertragungsaufgaben anzuzeigen.
4. Bei Endpunkten, für die bereits Übertragungsaufgaben vorhanden sind, werden diese im Abschnitt Übertragungsaufgaben angezeigt. Sie können wählen, ob Sie Details zu jeder Übertragungsaufgabe aus dieser Liste anzeigen möchten.

## Tutorial: Erste Schritte mit AWS Mainframe Modernization File Transfer

Mit AWS Mainframe Modernization File Transfer können Sie Mainframe-Datensätze für Anwendungsfälle wie Mainframe-Modernisierung, -Migration und Erweiterung übertragen und konvertieren.

Folgen Sie den Schritten in diesem Tutorial, um zu verstehen, wie AWS Mainframe Modernization File Transfer funktioniert.

### Übersicht

Die Dateiübertragung besteht aus den folgenden Komponenten:

1. Ein Agent, der auf dem Quell-Mainframe installiert werden soll.
2. Zugriff auf Funktionen zur Erkennung, Übertragung und Konvertierung von Datensätzen direkt über die AWS Mainframe Modernization Management Service Console.

Als Benutzer können Sie Datensätze vom Mainframe in Ihren Amazon S3 S3-Bucket übertragen.

### Themen

- [Schritt 1: Übertragen Sie das Tar-Paket der Agent-Binärdateien von AWS auf die logische Mainframe-Partition](#)
- [Schritt 2: Konfigurieren Sie den File Transfer Agent auf dem Quell-Mainframe](#)

- [Schritt 3: Erstellen Sie einen Datenübertragungsendpunkt](#)
- [Schritt 4: Erstellen Sie eine Übertragungsaufgabe](#)
- [Schritt 5: Status der Übertragungsaufgabe einsehen](#)

## Schritt 1: Übertragen Sie das Tar-Paket der Agent-Binärdateien von AWS auf die logische Mainframe-Partition

Laden Sie die Tar-Dateien über den Tar-Link für den [M2-Agent](#) herunter.

## Schritt 2: Konfigurieren Sie den File Transfer Agent auf dem Quell-Mainframe

In diesem Schritt konfigurieren und starten Sie den AWS Mainframe Modernization File Transfer Agent auf dem Quell-Mainframe. Der Agent ist erforderlich, um die Kommunikation zwischen der File Transfer-Servicefunktion und dem Quell-Mainframe zu ermöglichen. Pro Mainframe ist mindestens ein Agent erforderlich. Für hohe Verfügbarkeit und verbesserte Skalierbarkeit kann mehr als ein Agent gestartet werden.

Folgen Sie den Anweisungen im [the section called "Installieren Sie einen File Transfer Agent"](#) Handbuch, um die Installation des File Transfer Agents auf dem Mainframe abzuschließen.

## Schritt 3: Erstellen Sie einen Datenübertragungsendpunkt

Folgen Sie den Schritten auf [the section called "Endpunkte für die Datenübertragung"](#) Seite, um einen neuen Datenübertragungsendpunkt zu erstellen.

## Schritt 4: Erstellen Sie eine Übertragungsaufgabe

Folgen Sie den Schritten auf der [the section called "Aufgaben übertragen"](#) Seite, um Ihre Übertragungsaufgaben zu erstellen und zu verwalten.

## Schritt 5: Status der Übertragungsaufgabe einsehen

Sie können den Fortschritt Ihrer Übertragungsaufgabe in der AWS Mainframe-Modernisierungskonsole verfolgen. Weitere Einzelheiten finden Sie [the section called "Übertragungsaufgaben anzeigen"](#) im Abschnitt.

# Von AWS Mainframe Modernization File Transfer unterstützte Quell- und Zielkodierungen

AWS Mainframe Modernization File Transfer unterstützt verschiedene Datensatztypen und Codepage-Konvertierungsoptionen.

## Mainframe-Datensatztypen

AWS Mainframe Modernization File Transfer unterstützt die folgenden Mainframe-Datensatztypen:

- Nicht-VSAM: Sequentiell (PS), PDS, GDS, GDG
- VSAM-Typen: KSDS

## Unterstützte Codepages

AWS Mainframe Modernization File Transfer unterstützt die folgenden Codepages für die Konvertierung von Datensätzen (von/nach):

„BIG5“, „BIG5\_HKSCS“, „CESU\_8“, „EUC\_JP“, „EUC\_KR“, „GB18030“, „GB2312“, „GBK“, „IBM00858“, „IBM01140“, „IBM01141“, „IBM01142“, „IBM01142“, „IBM01143“, „IBM01144“, „IBM01144“ 145“, „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“, „IBM869“, „IBM870“, „IBM871“, „IBM918“, „IBM\_THAI“, „ISO\_2022\_CN“, „ISO\_2022\_JP“, „ISO\_2022\_JP\_2“, „ISO\_2022\_KR“, „ISO\_8859\_1“, „ISO\_8859\_13“, „ISO\_8859\_15“, „ISO\_8859\_16“, „ISO\_8859\_16“, „ISO\_8859\_2“, „ISO\_8859\_3“, „ISO\_8859\_4“, „ISO\_8859\_5“, „ISO\_8859\_6“, „ISO\_8859\_7“, „ISO\_8859\_8“, „ISO\_8859\_9“, „JIS\_X0201“, „JIS\_X0212\_1990“, „KOI8\_R“, „KOI8\_U“ FT\_JIS“, „TIS\_620“, „US\_ASCII“, „UTF\_16“, „UTF\_16BE“, „UTF\_16LE“, „UTF\_32“, „UTF\_32BE“, „UTF\_32BE“, „UTF\_32LE“, „UTF\_8“, „WINDOWS\_1250“, „WINDOWS\_1251“, „WINDOWS\_1252“, „WINDOWS\_1252“ 53“, „WINDOWS\_1254“, „WINDOWS\_1255“, „WINDOWS\_1256“, „WINDOWS\_1257“, „WINDOWS\_1258“, „WINDOWS\_31J“, „X\_BIG5\_HKSCS\_2001“, „X\_BIG5\_SOLARIS“, „X\_EUCJP\_OPEN“, „X\_EUC\_JP\_LINUX“, „X\_EUC\_TW“, „X\_IBM1006“, „X\_IBM1025“, „X\_IBM1046“, „X\_IBM1097“, „X\_IBM1098“, „X\_IBM1112“, „X\_IBM1122“, „X\_IBM1123“, „X\_IBM1124“, „X\_IBM1129“, „X\_IBM1166“, „X\_IBM1364“, „X\_IBM13831“, „X\_IBM1383“, „X\_IBM29626C“, „X\_IBM300“, „X\_IBM33722“, „X\_IBM737“, „X\_IBM833“, „X\_IBM834“, „X\_IBM856“, „X\_IBM874“, „X\_IBM875“, „X\_IBM921“, „X\_IBM921“, „X\_IBM921 IBM922“, „X\_IBM930“,



„X\_IBM933“, „X\_IBM935“, „X\_IBM937“, „X\_IBM939“, „X\_IBM942“, „X\_IBM 942“, „X\_IBM“  
942C“, „X\_IBM943“, „X\_IBM943C“, „X\_IBM948“, „X\_IBM949“, „X\_IBM949C“, „X\_IBM950“,  
„X\_IBM964“, „X\_IBM970“, „X\_ISCII91“, „X\_ISO\_2022\_22\_22\_CN\_CNS“, „X\_ISO\_2022\_CN\_GB“,  
„X\_ISO\_8859\_11“, „X\_JIS0208“, „X\_JISAUTODETECT“, „X\_JOHAB“, „X\_MACARABIC“,  
„X\_MACCENTRALEUROPE“, „X\_MACROATIAN“, „X\_MACCYRILLIC“, „X\_MACYRILLIC“ DINGBAT“,  
„X\_MACGREEK“, „X\_MACHEBREW“, „X\_MACICELAND“, „X\_MACROMAN“, „X\_MACROMANIA“,  
„X\_MACSYMBOL“, „X\_MACTHAI“, „X\_MACTURKISH“, „X\_MACUKRAINE“, „X\_MS932\_0213“,  
„X\_MS950\_HKSCS“, „X\_MS950\_HKSCS“, „X\_MS950\_HKSCS“\_HKSCS\_XP“, „X\_MSWIN\_936“,  
„X\_PCK“, „X\_SJIS\_0213“, „X\_UTF\_16LE\_BOM“, „X\_UTF\_32BE\_BOM“, „X\_UTF\_32LE\_BOM“,  
„X\_WINDOWS\_50220“, „X\_WINDOWS\_50221“, „X\_WINDOWS\_874“, „X\_WINDOWS\_949“,  
„X\_WINDOWS\_950“, „x\_Windows\_ISO2022J“

# Sicherheit bei der AWS Mainframe-Modernisierung

Cloud-Sicherheit hat AWS höchste Priorität. Als AWS Kunde profitieren Sie von einer Rechenzentrums- und Netzwerkarchitektur, die darauf ausgelegt sind, die Anforderungen der sicherheitssensibelsten Unternehmen zu erfüllen.

Sicherheit ist eine gemeinsame Verantwortung von Ihnen AWS und Ihnen. Das [Modell der übergreifenden Verantwortlichkeit](#) beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud — AWS ist verantwortlich für den Schutz der Infrastruktur, die AWS Dienste in der AWS Cloud ausführt. AWS bietet Ihnen auch Dienste, die Sie sicher nutzen können. Externe Prüfer testen und verifizieren regelmäßig die Wirksamkeit unserer Sicherheitsmaßnahmen im Rahmen der [AWS](#). Weitere Informationen zu den Compliance-Programmen, die für die AWS Mainframe-Modernisierung gelten, finden Sie unter [AWS-Services in Umfang nach Compliance-Programm](#).
- Sicherheit in der Cloud — Ihre Verantwortung richtet sich nach dem AWS Service, den Sie nutzen. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation hilft Ihnen zu verstehen, wie Sie das Modell der gemeinsamen Verantwortung bei der Nutzung der AWS Mainframe-Modernisierung anwenden können. Es zeigt Ihnen, wie Sie die AWS Mainframe-Modernisierung konfigurieren, um Ihre Sicherheits- und Compliance-Ziele zu erreichen. Sie erfahren auch, wie Sie andere AWS Services nutzen können, die Sie bei der Überwachung und Sicherung Ihrer AWS Mainframe-Modernisierungsressourcen unterstützen.

AWS Die Mainframe-Modernisierung stellt eigene IAM-geschützte Ressourcen (Anwendung, Umgebung, Bereitstellung usw.) bereit. Dabei handelt es sich um die administrativen Ressourcen für die AWS Mainframe-Modernisierung, für die jegliche Aktion gemäß den IAM-Richtlinien zulässig sein muss.

AWS Die Mainframe-Modernisierung zur Umstellung der Plattform wird ebenfalls durch IAM abgesichert. IAM erteilt oder verweigert einem Principal ebenfalls über Standard-IAM-Richtlinien die Erlaubnis für eine bestimmte Aktion an einer definierten Ressource, die von der ursprünglichen Mainframe-Umgebung abgeleitet wurde. Die AWS Mainframe Modernization Replatforming Runtime ruft den IAM-Autorisierungsdienst auf, wenn eine Anwendung versucht, eine solche Aktion auf

einer geschützten Ressource auszuführen. IAM gibt auf der Grundlage der standardmäßigen Bewertungsmechanismen für IAM-Richtlinien „Zulassen“ oder „Verweigern“ zurück.

## Inhalt

- [Datenschutz bei der AWS Mainframe-Modernisierung](#)
- [Identity and Access Management für die AWS Mainframe-Modernisierung](#)
- [Überprüfung der Einhaltung der Vorschriften für AWS die Mainframe-Modernisierung](#)
- [Resilienz bei der AWS Mainframe-Modernisierung](#)
- [Sicherheit der Infrastruktur in AWS Mainframe Modernization](#)
- [Zugriff AWS Mainframe Modernization über einen Schnittstellenendpunkt \(AWS PrivateLink\)](#)

## Datenschutz bei der AWS Mainframe-Modernisierung

Das [Modell der AWS gemeinsamen Verantwortung](#) gilt für den Datenschutz bei der AWS Mainframe-Modernisierung. Wie in diesem Modell beschrieben, AWS ist es für den Schutz der globalen Infrastruktur verantwortlich, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS-Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie unter [Häufig gestellte Fragen zum Datenschutz](#). Informationen zum Datenschutz in Europa finden Sie im Blog-Bertrag [AWS -Modell der geteilten Verantwortung und in der DSGVO](#) im AWS -Sicherheitsblog.

Aus Datenschutzgründen empfehlen wir, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto die Multi-Faktor-Authentifizierung (MFA).
- Verwenden Sie SSL/TLS, um mit Ressourcen zu kommunizieren. AWS Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein. AWS CloudTrail
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS-Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.

- Wenn Sie für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine API FIPS 140-2-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter [Federal Information Processing Standard \(FIPS\) 140-2](#).

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit AWS Mainframe Modernization oder anderen Programmen arbeiten und die Konsole, die AWS-Services API oder SDKs verwenden. AWS CLI AWS Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server bereitstellen, empfehlen wir dringend, keine Anmeldeinformationen zur Validierung Ihrer Anforderung an den betreffenden Server in die URL einzuschließen.

## Daten, die AWS Mainframe Modernization sammelt

AWS Mainframe Modernization sammelt verschiedene Arten von Daten von Ihnen:

- `Application configuration`: Dies ist eine JSON-Datei, die Sie erstellen, um Ihre Anwendung zu konfigurieren. Sie enthält Ihre Auswahlmöglichkeiten für die verschiedenen Optionen, die AWS Mainframe Modernization bietet. Die Datei enthält auch Informationen für abhängige AWS Ressourcen wie Amazon Simple Storage Service-Pfade, in denen Anwendungsartefakte gespeichert werden, oder den Amazon-Ressourcennamen (ARN), AWS Secrets Manager in dem Ihre Datenbankanmeldedaten gespeichert sind.
- `Application executable (binary)`: Dies ist eine Binärdatei, die Sie kompilieren und die Sie im Rahmen der AWS Mainframe-Modernisierung bereitstellen möchten.
- `Application JCL or scripts`: Dieser Quellcode verwaltet Batch-Jobs oder andere Verarbeitungen im Namen Ihrer Anwendung.
- `User application data`: Wenn Sie Datensätze importieren, speichert AWS Mainframe Modernization sie in der relationalen Datenbank, sodass Ihre Anwendung darauf zugreifen kann.
- `Application source code`: Mit Amazon AppStream 2.0 bietet AWS Mainframe Modernization eine Entwicklungsumgebung, in der Sie Code schreiben und kompilieren können.

AWS Mainframe Modernization speichert diese Daten nativ in AWS. Die Daten, die wir von Ihnen sammeln, werden in einem von der AWS Mainframe-Modernisierung verwalteten Amazon S3 S3-

Bucket gespeichert. Wenn Sie eine Anwendung bereitstellen, lädt AWS Mainframe Modernization die Daten auf eine Amazon Elastic Block Store-gestützte Amazon Elastic Compute Cloud-Instance herunter. Wenn die Bereinigung ausgelöst wird, werden die Daten aus dem Amazon EBS-Volume und aus Amazon S3 entfernt. Die Amazon EBS-Volumes sind Single-Tenant-Volumes, was bedeutet, dass eine Instance für einen Kunden verwendet wird. Instanzen werden niemals gemeinsam genutzt. Wenn Sie eine Laufzeitumgebung löschen, wird auch das Amazon EBS-Volume gelöscht. Wenn Sie eine Anwendung löschen, werden die Artefakte und die Konfiguration aus Amazon S3 gelöscht.

Anwendungsprotokolle werden in Amazon gespeichert CloudWatch. Protokollnachrichten von Kundenanwendungen werden CloudWatch ebenfalls nach exportiert. Die CloudWatch Protokolle können kundensensible Daten enthalten (z. B. Geschäftsdaten oder Sicherheitsinformationen in Debug-Meldungen). Weitere Informationen finden Sie unter [Überwachung der AWS Mainframe-Modernisierung mit Amazon CloudWatch](#).

Wenn Sie sich außerdem dafür entscheiden, ein oder mehrere Amazon Elastic File System- oder Amazon FSx-Dateisysteme an Ihre Laufzeitumgebung anzuhängen, werden die Daten in AWS diesen Systemen gespeichert. Sie müssen diese Daten bereinigen, wenn Sie die Dateisysteme nicht mehr verwenden möchten.

Sie können alle verfügbaren Amazon S3 S3-Verschlüsselungsoptionen verwenden, um Ihre Daten zu sichern, wenn Sie sie in den Amazon S3 S3-Bucket legen, den AWS Mainframe Modernization für die Anwendungsbereitstellung und den Import von Datensätzen verwendet. Darüber hinaus können Sie die Verschlüsselungsoptionen Amazon EFS und Amazon FSx verwenden, wenn Sie eines oder mehrere dieser Dateisysteme an Ihre Laufzeitumgebung anhängen.

## Datenverschlüsselung im Ruhezustand für den AWS Mainframe-Modernisierungsservice

AWS Mainframe Modernization lässt sich integrieren und bietet transparente serverseitige Verschlüsselung (SSE) für alle abhängigen Ressourcen, die Daten dauerhaft speichern, nämlich Amazon Simple Storage Service, Amazon DynamoDB und Amazon Elastic Block Store. AWS Key Management Service AWS Mainframe Modernization erstellt und verwaltet symmetrische AWS KMS Verschlüsselungsschlüssel für Sie in. AWS KMS


Die standardmäßige Verschlüsselung von Daten im Ruhezustand trägt dazu bei, den betrieblichen Aufwand und die Komplexität zu reduzieren, die mit dem Schutz vertraulicher Daten verbunden sind. Gleichzeitig ermöglicht es Ihnen die Migration von Anwendungen, für die strenge Verschlüsselungsvorschriften und gesetzliche Anforderungen gelten.

Sie können diese Verschlüsselungsebene nicht deaktivieren oder einen anderen Verschlüsselungstyp auswählen, wenn Sie Laufzeitumgebungen und Anwendungen erstellen.

Sie können Ihren eigenen, vom Kunden verwalteten Schlüssel für AWS Mainframe-Modernisierungsanwendungen und Laufzeitumgebungen verwenden, um Amazon S3- und Amazon EBS-Ressourcen zu verschlüsseln.

Für Ihre AWS Mainframe-Modernisierungsanwendungen können Sie diesen Schlüssel verwenden, um Ihre Anwendungsdefinition sowie andere Anwendungsressourcen wie JCL-Dateien zu verschlüsseln, die im Amazon S3 S3-Bucket gespeichert sind, der im Konto des Services erstellt wird. Weitere Informationen finden Sie unter [Erstellen einer Anwendung](#).

Für Ihre AWS Mainframe-Modernisierungs-Laufzeitumgebungen verwendet AWS Mainframe Modernization Ihren vom Kunden verwalteten Schlüssel, um das Amazon EBS-Volumen zu verschlüsseln, das es erstellt und an Ihre Amazon EC2 EC2-Instance für die AWS Mainframe-Modernisierung anhängt, das sich auch im Konto des Services befindet. Weitere Informationen finden Sie unter [Erstellen Sie eine Laufzeitumgebung](#).

 Note

DynamoDB-Ressourcen werden immer mit einem Dienstkonto Von AWS verwalteter Schlüssel im AWS Mainframe Modernization verschlüsselt. Sie können DynamoDB-Ressourcen nicht mit einem vom Kunden verwalteten Schlüssel verschlüsseln.

AWS Mainframe Modernization verwendet Ihren vom Kunden verwalteten Schlüssel für die folgenden Aufgaben:


- Eine Anwendung erneut bereitstellen.
- Ersetzen einer Amazon EC2 EC2-Instance für die AWS Mainframe-Modernisierung.

AWS Mainframe Modernization verwendet Ihren vom Kunden verwalteten Schlüssel nicht zur Verschlüsselung von Amazon Relational Database Service- oder Amazon Aurora Aurora-Datenbanken, Amazon Simple Queue Service-Warteschlangen und ElastiCache Amazon-Caches, die zur Unterstützung einer AWS Mainframe-Modernisierungsanwendung erstellt wurden, da keiner von ihnen Kundendaten enthält.

Weitere Informationen finden Sie unter [Kundenverwaltete Schlüssel](#) im AWS Key Management Service Entwicklerhandbuch.

Die folgende Tabelle fasst zusammen, wie AWS Mainframe Modernization Ihre sensiblen Daten verschlüsselt.

Datentyp	Von AWS verwalteter Schlüssel Verschlüsselung	Vom Kunden verwaltete Schlüsselverschlüsselung
<b>Definition</b> Enthält die Definition für eine bestimmte Anwendung.	Aktiviert	Aktiviert
<b>EnvironmentSummary</b> Enthält Informationen über die Laufzeitumgebung.	Aktiviert	Aktiviert
<b>ApplicationSummary</b> Enthält Informationen über die AWS Mainframe-Modernisierungsanwendung.	Aktiviert	Aktiviert
<b>DeploymentSummary</b> Enthält Informationen zur Bereitstellung einer AWS Mainframe-Modernisierungsanwendung.	Aktiviert	Aktiviert

 Note

AWS Die Mainframe-Modernisierung ermöglicht automatisch die Verschlüsselung im Ruhezustand Von AWS verwaltete Schlüssel , sodass Ihre vertraulichen Daten kostenlos geschützt werden. Für die Verwendung eines vom Kunden verwalteten Schlüssels fallen jedoch AWS KMS Gebühren an. Weitere Informationen über die Preise finden Sie unter [AWS Key Management Service – Preise](#).

Weitere Informationen zu finden AWS KMS Sie unter AWS Key Management Service.

## So verwendet AWS Mainframe Modernization Zuschüsse in AWS KMS

AWS Für die Mainframe-Modernisierung ist ein [Zuschuss](#) für die Nutzung Ihres vom Kunden verwalteten Schlüssels erforderlich.

Wenn Sie eine Anwendung oder Laufzeitumgebung erstellen oder eine Anwendung in AWS Mainframe Modernization bereitstellen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt ist, gewährt AWS Mainframe Modernization in Ihrem Namen einen Zuschuss, indem es eine [CreateGrant](#)Anfrage an sendet. AWS KMS Grants in AWS KMS werden verwendet, um AWS Mainframe Modernization Zugriff auf einen KMS-Schlüssel in einem Kundenkonto zu gewähren.

AWS Für die Mainframe-Modernisierung müssen Sie mit dem Zuschuss Ihren vom Kunden verwalteten Schlüssel für die folgenden internen Abläufe verwenden:

- Senden Sie [DescribeKey](#)Anfragen an, AWS KMS um zu überprüfen, ob die symmetrische, vom Kunden verwaltete Schlüssel-ID, die Sie bei der Erstellung einer Anwendung, Laufzeitumgebung oder Anwendungsbereitstellung eingegeben haben, gültig ist.
- Senden Sie [GenerateDataSchlüsselanfragen](#) an, AWS KMS um das Amazon EBS-Volume zu verschlüsseln, das Amazon EC2 EC2-Instances zugeordnet ist, die AWS Mainframe-Modernisierungs-Laufzeitumgebungen hosten.
- Senden Sie [Entschlüsselungsanforderungen](#) an, AWS KMS um verschlüsselte Inhalte auf Amazon EBS zu entschlüsseln.

AWS Mainframe Modernization verwendet AWS KMS Zuschüsse, um Ihre in Secrets Manager gespeicherten Geheimnisse zu entschlüsseln und wenn Sie eine Laufzeitumgebung erstellen, eine Anwendung erstellen oder erneut bereitstellen und eine Bereitstellung erstellen. Die Zuschüsse, die AWS Mainframe Modernization gewährt, unterstützen die folgenden Operationen:

- Erstellen oder aktualisieren Sie einen Zuschuss für eine Laufzeitumgebung:
  - Decrypt
  - Encrypt
  - ReEncryptVon
  - ReEncryptZu
  - GenerateDataSchlüssel
  - DescribeKey



- CreateGrant
- Einen Anwendungszuschuss erstellen oder erneut bereitstellen:
  - GenerateDataSchlüssel
- Erstellen Sie einen Bereitstellungszuschuss:
  - Decrypt

Sie können den Zugriff auf die Genehmigung jederzeit widerrufen oder den Zugriff des Services auf den vom Kunden verwalteten Schlüssel entfernen. Wenn Sie dies tun, kann AWS Mainframe Modernization auf keine Daten zugreifen, die mit dem vom Kunden verwalteten Schlüssel verschlüsselt wurden, was sich auf Vorgänge auswirkt, die von den Daten abhängen. Wenn die AWS Mainframe-Modernisierung beispielsweise versuchen würde, auf eine Anwendungsdefinition zuzugreifen, die mit einem vom Kunden verwalteten Schlüssel verschlüsselt wurde, ohne dass dieser Schlüssel gewährt wurde, würde die Anwendungserstellung fehlschlagen.

AWS Mainframe Modernization sammelt Benutzeranwendungskonfigurationen (JSON-Dateien) und Artefakte (Binärdateien und ausführbare Dateien). Es erstellt auch Metadaten, die verschiedene Entitäten verfolgen, die für den Betrieb der AWS Mainframe-Modernisierung verwendet werden, und erstellt Protokolle und Metriken. Zu den Protokollen und Metriken, die für Kunden sichtbar sind, gehören:

- CloudWatch Protokolle, die die Anwendung und die Runtime-Engine (entweder AWS Blu Age oder Micro Focus) widerspiegeln.
- CloudWatch Metriken für Betriebs-Dashboards.

Darüber hinaus sammelt AWS Mainframe Modernization Nutzungsdaten und Metriken für die Messung, Aktivitätsberichte usw. zu den Services. Diese Daten sind für Kunden nicht sichtbar.

AWS Mainframe Modernization speichert diese Daten je nach Datentyp an unterschiedlichen Orten. Kundendaten, die Sie hochladen, werden in einem Amazon S3 S3-Bucket gespeichert. Servicedaten werden sowohl in Amazon S3 als auch in DynamoDB gespeichert. Wenn Sie eine Anwendung bereitstellen, werden sowohl Ihre Daten als auch Ihre Servicedaten auf Amazon EBS-Volumes heruntergeladen. Wenn Sie Amazon EFS- oder Amazon FSx-Speicher an Ihre Laufzeitumgebung anhängen, werden die in diesen Dateisystemen gespeicherten Daten auch auf das Amazon EBS-Volume heruntergeladen.

Verschlüsselung im Ruhezustand ist standardmäßig konfiguriert. Sie können es nicht deaktivieren oder ändern. Derzeit können Sie auch die Konfiguration nicht ändern.

## Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel mithilfe der APIs AWS Management Console oder der AWS KMS APIs erstellen.

Einen symmetrischen kundenverwalteten Schlüssel erstellen

Folgen Sie den Schritten zum [Erstellen eines symmetrischen kundenverwalteten Schlüssels](#) im Entwicklerhandbuch zum AWS Key Management Service .

### Schlüsselrichtlinie

Schlüsselrichtlinien steuern den Zugriff auf den vom Kunden verwalteten Schlüssel. Jeder vom Kunden verwaltete Schlüssel muss über genau eine Schlüsselrichtlinie verfügen, die aussagt, wer den Schlüssel wie verwenden kann. Wenn Sie Ihren vom Kunden verwalteten Schlüssel erstellen, können Sie eine Schlüsselrichtlinie angeben. Weitere Informationen finden Sie unter [Verwalten des Zugriffs auf kundenverwaltete Schlüssel](#) im Entwicklerhandbuch zum AWS Key Management Service .

Um Ihren vom Kunden verwalteten Schlüssel mit Ihren AWS Mainframe-Modernisierungsressourcen zu verwenden, müssen die folgenden API-Operationen in der Schlüsselrichtlinie zulässig sein:

- [kms:CreateGrant](#): Fügt einem kundenverwalteten Schlüssel eine Erteilung hinzu. Gewährt Kontrollzugriff auf einen bestimmten KMS-Schlüssel, der den Zugriff auf [Grant-Operationen](#) ermöglicht, die für die AWS Mainframe-Modernisierung erforderlich sind. Weitere Informationen zur [Verwendung von Grants](#) finden Sie im AWS Key Management Service Developer Guide.

Auf diese Weise kann AWS Mainframe Modernization Folgendes tun:

- `GenerateDataKey` aufrufen, um einen verschlüsselten Datenschlüssel zu generieren und zu speichern, da der Datenschlüssel nicht sofort zum Verschlüsseln verwendet wird.
- `Decrypt` aufrufen, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf verschlüsselte Daten zu verwenden.
- Richten Sie einen Principal ein, der in den Ruhezustand geht, damit der Service `RetireGrant`
- [kms:DescribeKey](#)— Stellt dem Kunden verwaltete Schlüsselinformationen zur Verfügung, damit AWS Mainframe Modernization den Schlüssel validieren kann.

AWS Für die Mainframe-Modernisierung sind in den wichtigsten Richtlinien des Kunden die `kms:DescribeKey` entsprechenden Berechtigungen erforderlich `kms:CreateGrant`. AWS Die Mainframe-Modernisierung verwendet diese Richtlinie, um sich selbst einen Zuschuss zu gewähren.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "Enable IAM User Permissions",
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::AccountId:role/ExampleRole"
    },
    "Action": [
      "kms:CreateGrant",
      "kms:DescribeKey"
    ],
    "Resource": "*"
  }]
}
```

#### Note

Die Rolle, für die `Principal` im vorherigen Beispiel gezeigt wurde, ist die, die Sie für AWS Mainframe-Modernisierungsoperationen wie `CreateApplication` und verwenden. `CreateEnvironment`

Weitere Informationen zum [Festlegen von Berechtigungen in einer Richtlinie](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

Weitere Informationen zur [Fehlerbehebung beim Schlüsselzugriff](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.

## Angabe eines vom Kunden verwalteten Schlüssels für die AWS Mainframe-Modernisierung

Sie können einen vom Kunden verwalteten Schlüssel für die folgenden Ressourcen angeben:

- Anwendung
- Umgebung

Wenn Sie eine Ressource erstellen, können Sie den Schlüssel angeben, indem Sie eine KMS-ID eingeben, die AWS Mainframe Modernization verwendet, um die auf der Ressource gespeicherten sensiblen Daten zu verschlüsseln.

- KMS-ID — Eine [Schlüssel-ID](#) für einen vom Kunden verwalteten Schlüssel. Geben Sie eine Schlüssel-ID, einen Schlüssel-ARN, einen Alias-Namen oder einen Alias-ARN ein.

Sie können einen vom Kunden verwalteten Schlüssel mit dem AWS Management Console oder dem angeben AWS CLI.

Informationen zur Angabe Ihres vom Kunden verwalteten Schlüssels beim Erstellen einer Laufzeitumgebung in der AWS Management Console finden Sie unter [Erstellen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung](#). Informationen zur Angabe Ihres vom Kunden verwalteten Schlüssels bei der Erstellung einer Anwendung in der AWS Management Console finden Sie unter [Erstellen Sie eine Mainframe-Modernisierungsanwendung AWS](#).

Um Ihren vom Kunden verwalteten Schlüssel hinzuzufügen, wenn Sie eine Laufzeitumgebung mit dem erstellen AWS CLI, geben Sie den `kms-key-id` Parameter wie folgt an:

```
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
```

Um Ihren vom Kunden verwalteten Schlüssel hinzuzufügen, wenn Sie eine Anwendung mit dem erstellen AWS CLI, geben Sie den `kms-key-id` Parameter wie folgt an:

```
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 Verschlüsselungskontext für die Mainframe-Modernisierung

Ein [Verschlüsselungskontext](#) ist ein optionaler Satz von Schlüssel-Wert-Paaren, die zusätzliche kontextbezogene Informationen zu den Daten enthalten.

AWS KMS verwendet den Verschlüsselungskontext als [zusätzliche authentifizierte Daten, um die authentifizierte](#) Verschlüsselung zu unterstützen. Wenn Sie einen Verschlüsselungskontext in eine Anforderung zum Verschlüsseln von Daten einbeziehen, wird der Verschlüsselungskontext AWS

KMS an die verschlüsselten Daten gebunden. Zur Entschlüsselung von Daten müssen Sie denselben Verschlüsselungskontext in der Anfrage übergeben.

### AWS Verschlüsselungskontext für die Mainframe-Modernisierung

AWS Bei der Mainframe-Modernisierung wird bei allen AWS KMS kryptografischen Vorgängen im Zusammenhang mit einer Anwendung (Anwendung erstellen und Bereitstellung erstellen) derselbe Verschlüsselungskontext verwendet, wobei der Schlüssel `aws:m2:app` und der Wert die eindeutige Kennung der Anwendung ist.

### Example

```
"encryptionContextSubset": {
  "aws:m2:app": "a1bc2defabc3defabc4defabcd"
}
```

### Verwenden des Verschlüsselungskontexts für die Überwachung

Wenn Sie einen symmetrischen, vom Kunden verwalteten Schlüssel verwenden, um Ihre Anwendungen oder Laufzeitumgebungen zu verschlüsseln, können Sie den Verschlüsselungskontext auch in Prüfaufzeichnungen und Protokollen verwenden, um zu ermitteln, wie der vom Kunden verwaltete Schlüssel verwendet wird.

### Verwendung des Verschlüsselungskontextes zur Steuerung des Zugriffs auf den vom Kunden verwalteten Schlüssel

Sie können den Verschlüsselungskontext in Schlüsselrichtlinien und IAM-Richtlinien als `conditions` verwenden, um den Zugriff auf Ihren symmetrischen, kundenverwalteten Schlüssel zu kontrollieren. Sie können Verschlüsselungskontext-Einschränkungen auch in einer Genehmigung verwenden.

AWS Bei der Mainframe-Modernisierung wird der Zugriff auf den vom Kunden verwalteten Schlüssel in Ihrem Konto oder in Ihrer Region mithilfe von Einschränkungen durch den Verschlüsselungskontext bei Zuschüssen kontrolliert. Eine Genehmigungseinschränkung erfordert, dass durch die Genehmigung ermöglichte Vorgänge den angegebenen Verschlüsselungskontext verwenden. Das folgende Beispiel ist ein Zuschuss, den AWS Mainframe Modernization nutzt, um Anwendungsartefakte bei der Erstellung einer Anwendung zu verschlüsseln.

```
//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"  
  }  
}  
}
```

## Überwachung Ihrer Verschlüsselungsschlüssel für die Mainframe-Modernisierung AWS

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel mit Ihren AWS Mainframe-Modernisierungsressourcen verwenden, können Sie [Amazon CloudWatch Logs](#) verwenden [AWS CloudTrail](#), um Anfragen zu verfolgen, an die AWS Mainframe Modernization sendet. AWS KMS

### Beispiele für Laufzeitumgebungen

Die folgenden Beispiele sind AWS CloudTrail Ereignisse für `DescribeKey`, `CreateGrant`, und `Decrypt` zur Überwachung von KMS-Vorgängen `GenerateDataKey`, die von AWS Mainframe Modernization aufgerufen wurden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

#### DescribeKey

AWS Die Mainframe-Modernisierung verwendet den `DescribeKey` Vorgang, um zu überprüfen, ob der mit Ihrer Laufzeitumgebung verknüpfte vom AWS KMS Kunden verwaltete Schlüssel in dem Konto und in der Region vorhanden ist.

Das folgende Beispielergebnis zeichnet den Vorgang `DescribeKey` auf:

```
{  
  "eventVersion": "1.08",  
  "userIdentity": {  
    "type": "AssumedRole",  
    "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",  
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",  
    "accountId": "111122223333",  
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",  
    "sessionContext": {  
      "sessionIssuer": {  
        "type": "Role",
```

```
        "principalId": "AROAIKDTESTANDEXAMPLE: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

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel zur Verschlüsselung Ihrer Laufzeitumgebung verwenden, sendet AWS Mainframe Modernization in Ihrem Namen mehrere CreateGrant Anfragen, um die erforderlichen KMS-Operationen durchzuführen. Einige der Zuschüsse, die durch AWS Mainframe Modernization gewährt werden, werden sofort nach ihrer Nutzung zurückgezogen. Andere werden zurückgezogen, wenn Sie die Laufzeitumgebung löschen.

Das folgende Beispiereignis zeichnet den CreateGrant Vorgang für die Lambda-Ausführungsrolle auf, die dem Workflow Create Environment zugeordnet ist.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T20:11:45Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T20:23:09Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
```



```

    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
      "operations": [
        "Encrypt",
        "Decrypt",
        "ReEncryptFrom",
        "ReEncryptTo",
        "GenerateDataKey",
        "GenerateDataKey",
        "DescribeKey",
        "CreateGrant"
      ],
      "granteePrincipal": "m2.us-west-2.amazonaws.com",
      "retiringPrincipal": "m2.us-west-2.amazonaws.com"
    },
    "responseElements": {
      "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
      {
        "accountId": "111122223333",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
  }
}

```

Das folgende Beispiereignis zeichnet den CreateGrant Vorgang für die mit dem Dienst verknüpfte Auto Scaling Scaling-Gruppenrolle auf. Die Lambda-Ausführungsrolle, die dem Workflow Create Environment zugeordnet ist, ruft diesen CreateGrant Vorgang auf. Es erteilt

der Ausführungsrolle die Erlaubnis, einen Unterzuschuss für die serviceverknüpfte Rolle der Auto Scaling Scaling-Gruppe zu erstellen.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA3YPCLM65MZFPUM4J0: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

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für Ihre Laufzeitumgebungsressource aktivieren, erstellt Auto Scaling einen eindeutigen Schlüssel für die Verschlüsselung des Amazon EBS-Volumes, das der Laufzeitumgebung zugeordnet ist. Es sendet eine GenerateDataKey Anfrage an AWS KMS , in der der vom AWS KMS Kunden verwaltete Schlüssel für die Ressource angegeben ist.

Das folgende Beispiereignis zeichnet den Vorgang GenerateDataKey auf:

```
{
  "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

Wenn Sie auf eine verschlüsselte Laufzeitumgebung zugreifen, ruft Amazon EBS den Decrypt Vorgang auf, um den gespeicherten verschlüsselten Datenschlüssel für den Zugriff auf die verschlüsselten Daten zu verwenden.

Das folgende Beispiereignis zeichnet den Vorgang Decrypt auf:

```

{
  "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"
}

```

## Beispiele für Anwendungen

Bei den folgenden Beispielen handelt es sich um AWS CloudTrail Ereignisse für CreateGrant und GenerateDataKey zur Überwachung von KMS-Vorgängen, die von AWS Mainframe Modernization aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

### CreateGrant

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel zum Verschlüsseln Ihrer Anwendungsressourcen verwenden, sendet die Lambda-Ausführungsrolle in Ihrem Namen eine CreateGrant Anfrage, um auf den KMS-Schlüssel in Ihrem AWS Konto zuzugreifen. Der Zuschuss ermöglicht es der Lambda-Ausführungsrolle, Kundenanwendungsressourcen mithilfe

Ihres vom Kunden verwalteten Schlüssels auf Amazon S3 hochzuladen. Dieser Zuschuss wird unmittelbar nach der Erstellung der Anwendung zurückgezogen.

Das folgende Beispiereignis zeichnet den Vorgang CreateGrant auf:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T21:51:45Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T22:47:04Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "m2.us-west-2.amazonaws.com",
  "userAgent": "m2.us-west-2.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "constraints": {
      "encryptionContextSubset": {
        "aws:m2:app": "a1bc2defabc3defabc4defabcd"
      }
    }
  },
  "retiringPrincipal": "m2.us-west-2.amazonaws.com",
```

```

    "operations": [
      "GenerateDataKey"
    ],
    "granteePrincipal": "m2.us-west-2.amazonaws.com"
  },
  "responseElements": {
    "grantId":
"0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  },
  "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "111122223333",
  "eventCategory": "Management"
}

```

## GenerateDataKey

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel für Ihre Anwendungsressource aktivieren, erstellt die Lambda-Ausführungsrolle einen Schlüssel, mit dem Kundendaten verschlüsselt und in Amazon Simple Storage Service hochgeladen werden. Die Lambda-Ausführungsrolle sendet eine GenerateDataKey Anfrage an AWS KMS , die den vom AWS KMS Kunden verwalteten Schlüssel für die Ressource angibt.

Das folgende Beispiereignis zeichnet den Vorgang GenerateDataKey auf:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",

```



```
"principalId": "ARO3YPCLM65CLCEKKC7Z:ApplicationWorkflow-alpha-CreateApplicationVersion-CstWZUn5R4u6",
  "arn": "arn:aws:sts::111122223333:assumed-role/ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B/ApplicationWorkflow-alpha-CreateApplicationVersion-CstWZUn5R4u6",
  "accountId": "111122223333",
  "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
  "sessionContext": {
    "sessionIssuer": {
      "type": "Role",
      "principalId": "AROAIIGDTESTANDEXAMPLE:Sampleuser01",
      "arn": "arn:aws:iam::111122223333:role/ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B",
      "accountId": "111122223333",
      "userName": "ApplicationWorkflow-alpha-CreateApplicationVersion-1IZRBZYDG20B"
    },
    "webIdFederationData": {},
    "attributes": {
      "creationDate": "2022-12-06T23:28:32Z",
      "mfaAuthenticated": "false"
    }
  },
  "invokedBy": "m2.us-west-2.amazonaws.com"
},
"eventTime": "2022-12-06T23:29:08Z",
"eventSource": "kms.amazonaws.com",
"eventName": "GenerateDataKey",
"awsRegion": "us-west-2",
"sourceIPAddress": "m2.us-west-2.amazonaws.com",
"userAgent": "m2.us-west-2.amazonaws.com",
"requestParameters": {
  "encryptionContext": {
    "aws:m2:app": "a1bc2defabc3defabc4defabcd",
    "aws:s3:arn": "arn:aws:s3:::supernova-processedtemplate-111122223333-us-west-2/111122223333/a1bc2defabc3defabc4defabcd/1/cics-transaction/ZBNKE35.so"
  },
  "keySpec": "AES_256",
  "keyId": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
},
"responseElements": null,
"requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
"eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
```

```
"readOnly": true,
"resources": [
  {
    "accountId": "111122223333",
    "type": "AWS::KMS::Key",
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
  }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management"
}
```

## Beispiele für Bereitstellungen

Bei den folgenden Beispielen handelt es sich um AWS CloudTrail Ereignisse für `CreateGrant` und `Decrypt` zur Überwachung von KMS-Vorgängen, die von AWS Mainframe Modernization aufgerufen werden, um auf Daten zuzugreifen, die mit Ihrem vom Kunden verwalteten Schlüssel verschlüsselt wurden:

### CreateGrant

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel zur Verschlüsselung Ihrer Bereitstellungsressourcen verwenden, sendet AWS Mainframe Modernization in Ihrem Namen zwei `CreateGrant` Anfragen. Der erste Grant bezieht sich auf die aktuell `ListBatchJobScriptFiles` aufzurufende Lambda-Ausführungsrolle und wird sofort nach Abschluss der Bereitstellung zurückgezogen. Der zweite Zuschuss bezieht sich auf die Amazon EC2-Instance-Rolle mit eingeschränktem Geltungsbereich, sodass Amazon EC2 Kundenanwendungsressourcen von Amazon S3 herunterladen kann. Dieser Zuschuss wird zurückgezogen, wenn die Anwendung aus der Laufzeitumgebung gelöscht wird.

Das folgende Beispielergebnis zeichnet den Vorgang `CreateGrant` auf:

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
```

```

    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAIQDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-12-06T21:51:45Z",
        "mfaAuthenticated": "false"
      }
    },
    "invokedBy": "m2.us-west-2.amazonaws.com"
  },
  "eventTime": "2022-12-06T23:40:07Z",
  "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

Wenn Sie auf eine Bereitstellung zugreifen, ruft Amazon EC2 den Decrypt Vorgang auf, um den gespeicherten verschlüsselten Datenschlüssel zu verwenden, um verschlüsselte Kundendaten von Amazon S3 zu entschlüsseln und herunterzuladen.

Das folgende Beispiereignis zeichnet den Vorgang Decrypt auf:

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "ARO0A3YPCLM65BSPZ37E6G:m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "arn": "arn:aws:sts::111122223333:assumed-role/
SupernovaEnvironmentInstanceScopeDownRole/m2-hm-bqe367dxtfcpdbzmnhfzranisu",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "ARO0AIGDTESTANDEXAMPLE: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"
}
```

## Weitere Informationen

Die folgenden Ressourcen enthalten weitere Informationen zur Datenverschlüsselung im Ruhezustand:

- Weitere Informationen zu grundlegenden [AWS Key Management Service -Konzepten](#) finden Sie im AWS Key Management Service -Entwicklerhandbuch.
- Weitere Informationen finden Sie unter [Bewährte Methoden für die Sicherheit AWS Key Management Service](#) im AWS Key Management Service -Entwicklerhandbuch.

## Verschlüsselung während der Übertragung

Bei interaktiven Anwendungen, die Teil transaktionaler Workloads sind, wird der Datenaustausch zwischen dem Terminalemulator und dem AWS Mainframe-Modernisierungsdienst-Endpunkt für das TN3270-Protokoll während der Übertragung nicht verschlüsselt. Wenn für die Anwendung während der Übertragung eine Verschlüsselung erforderlich ist, sollten Sie möglicherweise einige zusätzliche Tunneling-Mechanismen implementieren.

AWS Die Mainframe-Modernisierung verwendet HTTPS, um die Service-APIs zu verschlüsseln. Die gesamte andere Kommunikation innerhalb der AWS Mainframe-Modernisierung wird durch die Service-VPC oder Sicherheitsgruppe sowie durch HTTPS geschützt. AWS Die Mainframe-Modernisierung überträgt Anwendungsartefakte, Konfigurationen und Anwendungsdaten. Anwendungsartefakte werden ebenso wie Anwendungsdaten aus einem Amazon S3 S3-Bucket kopiert, der Ihnen gehört. Sie können Anwendungskonfigurationen über einen Link zu Amazon S3 oder durch lokales Hochladen einer Datei bereitstellen.

Die grundlegende Verschlüsselung bei der Übertragung ist standardmäßig konfiguriert, gilt jedoch nicht für das TN3270-Protokoll. AWS Die Mainframe-Modernisierung verwendet HTTPS für API-Endpunkte, die ebenfalls standardmäßig konfiguriert sind.

## Identity and Access Management für die AWS Mainframe-Modernisierung

AWS Identity and Access Management (IAM) hilft einem Administrator AWS-Service , den Zugriff auf Ressourcen sicher zu kontrollieren. AWS IAM-Administratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um AWS Mainframe-

Modernisierungsressourcen zu verwenden. IAM ist ein Programm AWS-Service , das Sie ohne zusätzliche Kosten nutzen können.

## Themen

- [Zielgruppe](#)
- [Authentifizierung mit Identitäten](#)
- [Verwalten des Zugriffs mit Richtlinien](#)
- [So funktioniert die AWS Mainframe-Modernisierung mit IAM](#)
- [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)
- [Fehlerbehebung bei Identität und Zugriff bei der AWS Mainframe-Modernisierung](#)
- [Verwenden von serviceverknüpften Rollen für die Mainframe-Modernisierung](#)

## Zielgruppe

Die Art und Weise, wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie im Rahmen der AWS Mainframe-Modernisierung ausführen.

**Servicebenutzer** — Wenn Sie den AWS Mainframe Modernization Service für Ihre Arbeit verwenden, stellt Ihnen Ihr Administrator die erforderlichen Anmeldeinformationen und Berechtigungen zur Verfügung. Wenn Sie für Ihre Arbeit mehr Funktionen der AWS Mainframe-Modernisierung verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Funktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anfordern müssen. Wenn Sie in der AWS Mainframe-Modernisierung nicht auf eine Funktion zugreifen können, finden Sie weitere Informationen unter [Fehlerbehebung bei Identität und Zugriff bei der AWS Mainframe-Modernisierung](#)

**Serviceadministrator** — Wenn Sie in Ihrem Unternehmen für die Ressourcen zur AWS Mainframe-Modernisierung verantwortlich sind, haben Sie wahrscheinlich vollen Zugriff auf die AWS Mainframe-Modernisierung. Es ist Ihre Aufgabe, zu bestimmen, auf welche Funktionen und Ressourcen der AWS Mainframe-Modernisierung Ihre Servicebenutzer zugreifen sollen. Sie müssen dann Anträge an Ihren IAM-Administrator stellen, um die Berechtigungen Ihrer Servicenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die Grundkonzepte von IAM nachzuvollziehen. Weitere Informationen darüber, wie Ihr Unternehmen IAM mit der AWS Mainframe-Modernisierung nutzen kann, finden Sie unter [So funktioniert die AWS Mainframe-Modernisierung mit IAM](#)

**IAM-Administrator** — Wenn Sie ein IAM-Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien schreiben können, um den Zugriff auf die Mainframe-Modernisierung

zu verwalten. AWS Beispiele für identitätsbasierte Richtlinien zur AWS Mainframe-Modernisierung, die Sie in IAM verwenden können, finden Sie unter [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)

## Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen als IAM-Benutzer authentifiziert (angemeldet AWS) sein oder eine IAM-Rolle annehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAM Identity Center) -Benutzer, die Single Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als Verbundidentität anmelden, hat der Administrator vorher mithilfe von IAM-Rollen einen Identitätsverbund eingerichtet. Wenn Sie über den Verbund darauf zugreifen AWS, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter [So melden Sie sich bei Ihrem an AWS-Konto](#) im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit denen Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu [signieren, finden Sie im IAM-Benutzerhandbuch unter AWS API-Anfragen](#) signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter [Multi-Faktor-Authentifizierung](#) im AWS IAM Identity Center - Benutzerhandbuch und [Verwenden der Multi-Faktor-Authentifizierung \(MFA\) in AWS](#) im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie ein neues AWS-Konto erstellen, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS-Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-



Benutzer bezeichnet. Der Zugriff erfolgt, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie unter [Aufgaben, die Root-Benutzer-Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

## Verbundidentität

Als bewährte Methode sollten menschliche Benutzer, einschließlich Benutzer, die Administratorzugriff benötigen, für den Zugriff AWS-Services mithilfe temporärer Anmeldeinformationen den Verbund mit einem Identitätsanbieter verwenden.

Eine föderierte Identität ist ein Benutzer aus Ihrem Unternehmensbenutzerverzeichnis, einem Web-Identitätsanbieter AWS Directory Service, dem Identity Center-Verzeichnis oder einem beliebigen Benutzer, der mithilfe AWS-Services von Anmeldeinformationen zugreift, die über eine Identitätsquelle bereitgestellt wurden. Wenn föderierte Identitäten darauf zugreifen AWS-Konten, übernehmen sie Rollen, und die Rollen stellen temporäre Anmeldeinformationen bereit.

Für die zentrale Zugriffsverwaltung empfehlen wir Ihnen, AWS IAM Identity Center zu verwenden. Sie können Benutzer und Gruppen in IAM Identity Center erstellen, oder Sie können eine Verbindung zu einer Gruppe von Benutzern und Gruppen in Ihrer eigenen Identitätsquelle herstellen und diese synchronisieren, um sie in all Ihren AWS-Konten Anwendungen zu verwenden. Informationen zu IAM Identity Center finden Sie unter [Was ist IAM Identity Center?](#) im AWS IAM Identity Center - Benutzerhandbuch.

## IAM-Benutzer und -Gruppen

Ein [IAM-Benutzer](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wenn möglich, empfehlen wir, temporäre Anmeldeinformationen zu verwenden, anstatt IAM-Benutzer zu erstellen, die langfristige Anmeldeinformationen wie Passwörter und Zugriffsschlüssel haben. Bei speziellen Anwendungsfällen, die langfristige Anmeldeinformationen mit IAM-Benutzern erfordern, empfehlen wir jedoch, die Zugriffsschlüssel zu rotieren. Weitere Informationen finden Sie unter [Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, die langfristige Anmeldeinformationen erfordern](#) im IAM-Benutzerhandbuch.

Eine [IAM-Gruppe](#) ist eine Identität, die eine Sammlung von IAM-Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer

gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise einer Gruppe mit dem Namen IAMAdmins Berechtigungen zum Verwalten von IAM-Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter [Erstellen eines IAM-Benutzers \(anstatt einer Rolle\)](#) im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine [IAM-Rolle](#) ist eine Identität innerhalb Ihres Unternehmens AWS-Konto, die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, ist aber nicht mit einer bestimmten Person verknüpft. Sie können vorübergehend eine IAM-Rolle in der übernehmen, AWS Management Console indem Sie die Rollen [wechseln](#). Sie können eine Rolle übernehmen, indem Sie eine AWS CLI oder AWS API-Operation aufrufen oder eine benutzerdefinierte URL verwenden. Weitere Informationen zu Methoden für die Verwendung von Rollen finden Sie unter [Verwenden von IAM-Rollen](#) im IAM-Benutzerhandbuch.

IAM-Rollen mit temporären Anmeldeinformationen sind in folgenden Situationen hilfreich:

- **Verbundbenutzerzugriff** – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie unter [Erstellen von Rollen für externe Identitätsanbieter](#) im IAM-Benutzerhandbuch. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Wenn Sie steuern möchten, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in IAM. Informationen zu Berechtigungssätzen finden Sie unter [Berechtigungssätze](#) im AWS IAM Identity Center -Benutzerhandbuch.
- **Temporäre IAM-Benutzerberechtigungen** – Ein IAM-Benutzer oder eine -Rolle kann eine IAM-Rolle übernehmen, um vorübergehend andere Berechtigungen für eine bestimmte Aufgabe zu erhalten.
- **Kontoübergreifender Zugriff** – Sie können eine IAM-Rolle verwenden, um einem vertrauenswürdigen Prinzipal in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS-Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zu den

Unterschieden zwischen Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

- **Serviceübergreifender Zugriff** — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise einen Aufruf in einem Service tätigen, führt dieser Service häufig Anwendungen in Amazon-EC2 aus oder speichert Objekte in Amazon-S3. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
- **Forward Access Sessions (FAS)** — Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, in Kombination mit der Anfrage, Anfragen an AWS-Service nachgelagerte Dienste zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).
- **Servicerolle** – Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.
- **Dienstbezogene Rolle** — Eine dienstbezogene Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.
- **Auf Amazon EC2 ausgeführte Anwendungen** — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2-Instance ausgeführt werden und API-Anfragen stellen AWS CLI . AWS Das ist eher zu empfehlen, als Zugriffsschlüssel innerhalb der EC2-Instance zu speichern. Um einer EC2-Instance eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein Instance-Profil, das an die Instance angehängt ist. Ein Instance-Profil enthält die Rolle und ermöglicht, dass Programme, die in der EC2-Instance ausgeführt werden, temporäre Anmeldeinformationen erhalten. Weitere Informationen finden Sie unter [Verwenden einer IAM-](#)

[Rolle zum Erteilen von Berechtigungen für Anwendungen, die auf Amazon-EC2-Instances ausgeführt werden](#) im IAM-Benutzerhandbuch.

Informationen dazu, wann Sie IAM-Rollen oder IAM-Benutzer verwenden sollten, finden Sie unter [Erstellen einer IAM-Rolle \(anstatt eines Benutzers\)](#) im IAM-Benutzerhandbuch.

## Verwalten des Zugriffs mit Richtlinien

Sie kontrollieren den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden AWS als JSON-Dokumente gespeichert. Weitere Informationen zu Struktur und Inhalten von JSON-Richtliniendokumenten finden Sie unter [Übersicht über JSON-Richtlinien](#) im IAM-Benutzerhandbuch.

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

IAM-Richtlinien definieren Berechtigungen für eine Aktion unabhängig von der Methode, die Sie zur Ausführung der Aktion verwenden. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen von der AWS Management Console AWS CLI, der oder der AWS API abrufen.

## Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können AWS-Konto. Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie und einer eingebundenen Richtlinie wählen, finden Sie unter [Auswahl zwischen verwalteten und eingebundenen Richtlinien](#) im IAM-Benutzerhandbuch.

## Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien von IAM nicht in einer ressourcenbasierten Richtlinie verwenden.

## Zugriffssteuerungslisten (ACLs)

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Services, die ACLs unterstützen. AWS WAF Weitere Informationen“ zu ACLs finden Sie unter [Zugriffskontrollliste \(ACL\) – Übersicht](#) (Access Control List) im Amazon-Simple-Storage-Service-Entwicklerhandbuch.

## Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- **Berechtigungsgrenzen** – Eine Berechtigungsgrenze ist ein erweitertes Feature, mit der Sie die maximalen Berechtigungen festlegen können, die eine identitätsbasierte Richtlinie einer IAM-Entität (IAM-Benutzer oder -Rolle) erteilen kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen über Berechtigungsgrenzen finden Sie unter [Berechtigungsgrenzen für IAM-Entitäten](#) im IAM-Benutzerhandbuch.
- **Service Control Policies (SCPs)** — SCPs sind JSON-Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen. AWS Organizations AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer Objekte AWS-Konten , die Ihrem Unternehmen gehören. Wenn Sie innerhalb einer Organisation alle Features aktivieren, können Sie Service-Kontrollrichtlinien (SCPs) auf alle oder einzelne Ihrer Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten. Root-Benutzer des AWS-Kontos Weitere Informationen zu Organizations und SCPs finden Sie unter [Funktionsweise von SCPs](#) im AWS Organizations -Benutzerhandbuch.
- **Sitzungsrichtlinien** – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie unter [Sitzungsrichtlinien](#) im IAM-Benutzerhandbuch.

## Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAM-Benutzerhandbuch unter [Bewertungslogik für Richtlinien](#).

## So funktioniert die AWS Mainframe-Modernisierung mit IAM

Bevor Sie IAM zur Verwaltung des Zugriffs auf die AWS Mainframe-Modernisierung verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen für die Mainframe-Modernisierung verfügbar sind. AWS

IAM-Funktionen, die Sie mit der Mainframe-Modernisierung verwenden können AWS

IAM-Feature	AWS Unterstützung für die Mainframe-Modernisierung
<a href="#">Identitätsbasierte Richtlinien</a>	Ja
<a href="#">Ressourcenbasierte Richtlinien</a>	Nein
<a href="#">Richtlinienaktionen</a>	Ja
<a href="#">Richtlinienressourcen</a>	Ja
<a href="#">Bedingungsschlüssel für die Richtlinie</a>	Ja
<a href="#">ACLs</a>	Nein
<a href="#">ABAC (Tags in Richtlinien)</a>	Ja
<a href="#">Temporäre Anmeldeinformationen</a>	Ja
<a href="#">Forward Access Sessions (FAS)</a>	Ja
<a href="#">Servicerollen</a>	Ja
<a href="#">Service-verknüpfte Rollen</a>	Ja

Einen allgemeinen Überblick darüber, wie die AWS Mainframe-Modernisierung und andere AWS Services mit den meisten IAM-Funktionen funktionieren, finden Sie im [AWS IAM-Benutzerhandbuch unter Services, die mit IAM funktionieren](#).

## Identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

Unterstützt Richtlinien auf Identitätsbasis. Ja

Identitätsbasierte Richtlinien sind JSON-Berechtigungsrichtliniendokumente, die Sie einer Identität anfügen können, wie z. B. IAM-Benutzern, -Benutzergruppen oder -Rollen. Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen identitätsbasierter Richtlinien finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Mit identitätsbasierten IAM-Richtlinien können Sie angeben, welche Aktionen und Ressourcen zugelassen oder abgelehnt werden. Darüber hinaus können Sie die Bedingungen festlegen, unter denen Aktionen zugelassen oder abgelehnt werden. Sie können den Prinzipal nicht in einer identitätsbasierten Richtlinie angeben, da er für den Benutzer oder die Rolle gilt, dem er zugeordnet ist. Informationen zu sämtlichen Elementen, die Sie in einer JSON-Richtlinie verwenden, finden Sie in der [IAM-Referenz für JSON-Richtlinienelemente](#) im IAM-Benutzerhandbuch.

### Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

Beispiele für identitätsbasierte Richtlinien zur AWS Mainframe-Modernisierung finden Sie unter [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)

## Ressourcenbasierte Richtlinien im Rahmen der Mainframe-Modernisierung AWS

Unterstützt ressourcenbasierte Richtlinien Nein

Ressourcenbasierte Richtlinien sind JSON-Richtliniendokumente, die Sie an eine Ressource anfügen. Beispiele für ressourcenbasierte Richtlinien sind IAM-Rollen-Vertrauensrichtlinien und Amazon-S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie [einen Prinzipal angeben](#). Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS-Services



Um kontoübergreifenden Zugriff zu ermöglichen, können Sie ein gesamtes Konto oder IAM-Entitäten in einem anderen Konto als Prinzipal in einer ressourcenbasierten Richtlinie angeben. Durch das Hinzufügen eines kontoübergreifenden Auftraggebers zu einer ressourcenbasierten Richtlinie ist nur die halbe Vertrauensbeziehung eingerichtet. Wenn sich der Prinzipal und die Ressource unterscheiden AWS-Konten, muss ein IAM-Administrator des vertrauenswürdigen Kontos auch der Prinzipalentität (Benutzer oder Rolle) die Berechtigung zum Zugriff auf die Ressource erteilen. Sie erteilen Berechtigungen, indem Sie der juristischen Stelle eine identitätsbasierte Richtlinie anfügen. Wenn jedoch eine ressourcenbasierte Richtlinie Zugriff auf einen Prinzipal in demselben Konto gewährt, ist keine zusätzliche identitätsbasierte Richtlinie erforderlich. Weitere Informationen finden Sie unter [Wie sich IAM-Rollen von ressourcenbasierten Richtlinien unterscheiden](#) im IAM-Benutzerhandbuch.

## Politische Maßnahmen für die AWS Mainframe-Modernisierung

Unterstützt Richtlinienaktionen

Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Action` einer JSON-Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API-Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, die nur mit Genehmigung durchgeführt werden können und für die es keinen passenden API-Vorgang gibt. Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Eine Liste der AWS Mainframe-Modernisierungsaktionen finden Sie unter [Durch die AWS Mainframe-Modernisierung definierte Aktionen](#) in der Serviceautorisierungsreferenz.

Bei Richtlinienaktionen bei der AWS Mainframe-Modernisierung wird vor der Aktion das folgende Präfix verwendet:

m2

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
  "m2:StartApplication",
  "m2:StopApplication"
]
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `List` beginnen, einschließlich der folgenden Aktion:

```
"Action": "m2:List*"
```

Beispiele für identitätsbasierte Richtlinien zur AWS Mainframe-Modernisierung finden Sie unter [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)

## Richtlinienressourcen für die Mainframe-Modernisierung AWS

Unterstützt Richtlinienressourcen

Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das bedeutet die Festlegung, welcher Prinzipal Aktionen für welche Ressourcen unter welchen Bedingungen ausführen kann.

Das JSON-Richtlinienelement `Resource` gibt die Objekte an, auf welche die Aktion angewendet wird. Anweisungen müssen entweder ein `Resource` oder ein `NotResource`-Element enthalten. Als bewährte Methode geben Sie eine Ressource mit dem zugehörigen [Amazon-Ressourcennamen \(ARN\)](#) an. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (\*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

```
"Resource": "*"
```

Sie können den Zugriff auf bestimmte Ressourcen für die AWS Mainframe-Modernisierung einschränken, indem Sie anhand ihrer ARNs die Ressource identifizieren, für die die IAM-Richtlinie

gilt. Weitere Informationen zum Format von ARNs finden Sie unter [Amazon-Ressourcennamen \(ARNs\)](#) im Allgemeine AWS-Referenz.

Eine AWS Mainframe-Modernisierungsumgebung hat beispielsweise den folgenden ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:env/service-generated-unique-identifier"
```

Eine AWS Mainframe-Modernisierungsanwendung hat den folgenden ARN.

```
"Resource": "arn:aws:m2:regionId:accountId:app/service-generated-unique-identifier"
```

Nicht alle AWS Mainframe-Modernisierungsaktionen unterstützen Berechtigungen auf Ressourcenebene. Für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, müssen Sie den Platzhalter (\*) verwenden.

Die folgenden AWS Mainframe-Modernisierungsaktionen unterstützen keine Berechtigungen auf Ressourcenebene.

```
ListApplications
  ListApplicationVersions
  ListBatchJobDefinitions
  ListBatchJobExecutions
  ListDataSetImportHistory
  ListDataSets
  ListDeployments
  ListEngineVersions
  ListEnvironments
  ListTagsForResource
```

Eine Liste der Ressourcentypen für die AWS Mainframe-Modernisierung und ihre ARNs finden Sie unter [Resources Defined by AWS Mainframe Modernization in der Service Authorization Reference](#). Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter [Durch die AWS Mainframe-Modernisierung definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien zur AWS Mainframe-Modernisierung finden Sie unter [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)

## AWS API-Berechtigungen für die Mainframe-Modernisierung: Referenz zu Aktionen, Ressourcen und Bedingungen

Wenn Sie Berechtigungsrichtlinien schreiben, die Sie an eine IAM-Identität anhängen können (identitätsbasierte Richtlinien), können Sie die folgende Tabelle als Referenz verwenden. Die Tabelle enthält Folgendes:

- Jeder API-Vorgang zur AWS Mainframe-Modernisierung
- Die entsprechenden Aktionen, für die Sie Berechtigungen zum Ausführen der Aktion erteilen können
- Die AWS Ressource, für die Sie die Berechtigungen erteilen können

Die Aktionen geben Sie im Feld `Action` und den Wert für die Ressource im Feld `Resource` der Richtlinie an.

Sie können AWS globale Bedingungsschlüssel in Ihren AWS Mainframe-Modernisierungsrichtlinien verwenden, um Bedingungen auszudrücken. Eine vollständige Liste der AWS Schlüssel finden Sie unter [Verfügbare globale Bedingungsschlüssel](#) im IAM-Benutzerhandbuch.

### Note

Um eine Aktion anzugeben, verwenden Sie das Präfix `m2:` gefolgt vom Namen der API-Operation (z. B. `m2:CreateApplication`).

### AWS Mainframe-Modernisierungs-API und erforderliche Berechtigungen für Aktionen

AWS API-Operationen zur Mainframe-Modernisierung	Erforderliche Berechtigungen (API-Aktionen)	Ressourcen
<a href="#">CancelBatchJobExecution</a>		Anwendung
<a href="#">CreateApplication</a>	<code>s3:GetObject</code> <code>s3:ListBucket</code>	Anwendung

AWS API-Operationen zur Mainframe-Modernisierung	Erforderliche Berechtigungen (API-Aktionen)	Ressourcen
<a href="#">CreateDataSetImportAufgabe</a>	m2:CreateDataSetImportTask s3:GetObject	Anwendung
<a href="#">CreateDeployment</a>	elasticloadbalancing:AddTags elasticloadbalancing:CreateListener elasticloadbalancing:CreateTargetGroup elasticloadbalancing:RegisterTargets	Anwendung

AWS API-Operationen zur Mainframe-Modernisierung	Erforderliche Berechtigungen (API-Aktionen)	Ressourcen
<a href="#">CreateEnvironment</a>	ec2:CreateNetworkInterface  ec2:CreateNetworkInterfacePermission  ec2:DescribeNetworkInterfaces  ec2:DescribeSecurityGroups  ec2:DescribeSubnets  ec2:DescribeVpcAttribute  ec2:DescribeVpcs  ec2:ModifyNetworkInterfaceAttribute  elasticfilesystem:DescribeMountTargets  elasticloadbalancing:AddTags  elasticloadbalancing:CreateLoadBalancer  fsx:DescribeFileSystems  iam:CreateServiceLinkedRole	Umgebung

AWS API-Operationen zur Mainframe-Modernisierung	Erforderliche Berechtigungen (API-Aktionen)	Ressourcen
<a href="#">DeleteApplication</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup  logs:DeleteLogDelivery	Anwendung
<a href="#">DeleteApplicationFromEnvironment</a>	elasticloadbalancing:DeleteListener  elasticloadbalancing:DeleteTargetGroup	Anwendung  Umgebung
<a href="#">DeleteEnvironment</a>	elasticloadbalancing:DeleteLoadBalancer	Umgebung
<a href="#">GetApplication</a>		Anwendung
<a href="#">GetApplicationVersion</a>		Anwendung
<a href="#">GetBatchJobExecution</a>		Anwendung
<a href="#">GetDataSetDetails</a>		Anwendung
<a href="#">GetDataSetImportAufgabe</a>		Anwendung
<a href="#">GetDeployment</a>		Anwendung
<a href="#">GetEnvironment</a>		Umgebung
<a href="#">ListApplications</a>		*
<a href="#">ListApplicationVersionen</a>		*

AWS API-Operationen zur Mainframe-Modernisierung	Erforderliche Berechtigungen (API-Aktionen)	Ressourcen
<a href="#">ListBatchJobDefinitions</a>		*
<a href="#">ListBatchJobExecutions</a>		*
<a href="#">ListDataSetImportGeschichte</a>		*
<a href="#">ListDataSets</a>		*
<a href="#">ListDeployments</a>		*
<a href="#">ListEngineVersionen</a>		*
<a href="#">ListEnvironments</a>		*
<a href="#">ListTagsForResource</a>		*
<a href="#">StartApplication</a>		Anwendung
<a href="#">StartBatchJob</a>		Anwendung
<a href="#">StopApplication</a>		Anwendung
<a href="#">TagResource</a>		*
<a href="#">UntagResource</a>		*
<a href="#">UpdateApplication</a>	s3:GetObject s3:ListBucket	Anwendung
<a href="#">UpdateEnvironment</a>		Umgebung



## Die wichtigsten politischen Bedingungen für die AWS Mainframe-Modernisierung

Unterstützt servicespezifische Richtlinienbedingungsschlüssel	Ja
---	----

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition block`) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die [Bedingungsoperatoren](#) verwenden, z. B. `ist gleich` oder `kleiner als`, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen AND-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, wertet die Bedingung mithilfe einer logischen OR Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Beispielsweise können Sie einem IAM-Benutzer die Berechtigung für den Zugriff auf eine Ressource nur dann gewähren, wenn sie mit dessen IAM-Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie unter [IAM-Richtlinienelemente: Variablen und Tags](#) im IAM-Benutzerhandbuch.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter [Kontextschlüssel für AWS globale Bedingungen](#) im IAM-Benutzerhandbuch.

Die folgenden Bedingungsschlüssel sind spezifisch für die AWS Mainframe-Modernisierung

```
m2:EngineType
m2:InstanceType
```

Eine Liste der Bedingungsschlüssel für die AWS Mainframe-Modernisierung finden Sie unter [Bedingungsschlüssel für die AWS Mainframe-Modernisierung](#) in der Service Authorization Reference. Informationen darüber, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter [Durch die AWS Mainframe-Modernisierung definierte Aktionen](#).

Beispiele für identitätsbasierte Richtlinien für die AWS Mainframe-Modernisierung finden Sie unter [Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS](#)

## Zugriffskontrolllisten (ACLs) bei der Mainframe-Modernisierung AWS

Unterstützt ACLs	Nein
------------------	------

Zugriffssteuerungslisten (ACLs) steuern, welche Prinzipale (Kontomitglieder, Benutzer oder Rollen) auf eine Ressource zugreifen können. ACLs sind ähnlich wie ressourcenbasierte Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

## Attributbasierte Zugriffskontrolle (ABAC) mit Mainframe-Modernisierung AWS

Unterstützt ABAC (Tags in Richtlinien)	Ja
--	----

Die attributbasierte Zugriffskontrolle (ABAC) ist eine Autorisierungsstrategie, bei der Berechtigungen basierend auf Attributen definiert werden. In werden diese AWS Attribute als Tags bezeichnet. Sie können Tags an IAM-Entitäten (Benutzer oder Rollen) und an viele AWS Ressourcen anhängen. Das Markieren von Entitäten und Ressourcen ist der erste Schritt von ABAC. Anschließend entwerfen Sie ABAC-Richtlinien, um Operationen zuzulassen, wenn das Tag des Prinzipals mit dem Tag der Ressource übereinstimmt, auf die sie zugreifen möchten.

ABAC ist in Umgebungen hilfreich, die schnell wachsen, und unterstützt Sie in Situationen, in denen die Richtlinienverwaltung mühsam wird.

Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungelement einer [Richtlinie Tag-Informationen](#) an, indem Sie die Schlüssel `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, oder Bedingung `aws:TagKeys` verwenden.

Wenn ein Service alle drei Bedingungsschlüssel für jeden Ressourcentyp unterstützt, lautet der Wert für den Service Ja. Wenn ein Service alle drei Bedingungsschlüssel für nur einige Ressourcentypen unterstützt, lautet der Wert Teilweise.

Weitere Informationen zu ABAC finden Sie unter [Was ist ABAC?](#) im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe [Attributbasierte Zugriffskontrolle \(ABAC\)](#) verwenden im IAM-Benutzerhandbuch.

## Temporäre Anmeldeinformationen bei der AWS Mainframe-Modernisierung verwenden

Unterstützt temporäre Anmeldeinformationen      Ja

Einige funktionieren AWS-Services nicht, wenn Sie sich mit temporären Anmeldeinformationen anmelden. Weitere Informationen, einschließlich Informationen, die mit temporären Anmeldeinformationen AWS-Services [funktionieren AWS-Services](#) , [finden Sie im IAM-Benutzerhandbuch unter Diese Option funktioniert mit IAM](#).

Sie verwenden temporäre Anmeldeinformationen, wenn Sie sich mit einer anderen AWS Management Console Methode als einem Benutzernamen und einem Passwort anmelden. Wenn Sie beispielsweise AWS über den Single Sign-On-Link (SSO) Ihres Unternehmens darauf zugreifen, werden bei diesem Vorgang automatisch temporäre Anmeldeinformationen erstellt. Sie erstellen auch automatisch temporäre Anmeldeinformationen, wenn Sie sich als Benutzer bei der Konsole anmelden und dann die Rollen wechseln. Weitere Informationen zum Wechseln von Rollen finden Sie unter [Wechseln zu einer Rolle \(Konsole\)](#) im IAM-Benutzerhandbuch.

Mithilfe der AWS API AWS CLI oder können Sie temporäre Anmeldeinformationen manuell erstellen. Sie können diese temporären Anmeldeinformationen dann für den Zugriff verwenden AWS. AWS empfiehlt, temporäre Anmeldeinformationen dynamisch zu generieren, anstatt langfristige Zugriffsschlüssel zu verwenden. Weitere Informationen finden Sie unter [Temporäre Sicherheitsanmeldeinformationen in IAM](#).

## Zugriffssitzungen für die AWS Mainframe-Modernisierung weiterleiten

Unterstützt Forward Access Sessions (FAS)      Ja

Wenn Sie einen IAM-Benutzer oder eine IAM-Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FAS verwendet die Berechtigungen des Prinzipals, der einen aufruft AWS-Service, kombiniert mit der Anforderung, Anfragen an nachgelagerte Dienste AWS-Service zu stellen. FAS-Anfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS-Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien für FAS-Anfragen finden Sie unter [Zugriffssitzungen weiterleiten](#).

**⚠ Important**

Diese Token gewähren AWS Mainframe Modernization ohne Ihre ausdrückliche Zustimmung Zugriff auf Kundendaten. AWS Mainframe Modernization stellt beispielsweise Anwendungsartefakte mit zugehörigen Geschäftsdaten aus einem Amazon S3 S3-Bucket bereit, ohne die ausdrückliche Genehmigung des Kunden einzuholen. Möglicherweise müssen Sie alle Compliance-Unterlagen entsprechend aktualisieren.

## Servicerollen für die AWS Mainframe-Modernisierung

Unterstützt Servicerollen

Ja

Eine Servicerolle ist eine [IAM-Rolle](#), die ein Service annimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM-Administrator kann eine Servicerolle innerhalb von IAM erstellen, ändern und löschen. Weitere Informationen finden Sie unter [Erstellen einer Rolle zum Delegieren von Berechtigungen an einen AWS-Service](#) im IAM-Benutzerhandbuch.

AWS Die Mainframe-Modernisierung unterstützt Servicerollen für Aktivitäts-Hooks (Transaktion/Jobs am Ende oder Abschluss usw.).

**⚠ Warning**

Durch das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität der AWS Mainframe-Modernisierung beeinträchtigt werden. Bearbeiten Sie Servicerollen nur, wenn AWS Mainframe Modernization Sie dazu anleitet.

## Auswahl einer IAM-Rolle bei der Mainframe-Modernisierung AWS

Wenn Sie zuvor eine IAM-Rolle erstellt haben, die Ihre auf Amazon EC2 ausgeführten Anwendungen annehmen können, können Sie diese Rolle wählen, wenn Sie eine Startvorlage oder Startkonfiguration erstellen. AWS Mainframe Modernization bietet Ihnen eine Liste von Rollen, aus denen Sie wählen können. Beim Erstellen dieser Rollen ist es wichtig, IAM-Richtlinien mit geringsten Berechtigungen zuzuordnen, die den Zugriff auf die spezifischen API-Aufrufe einschränken, die die Anwendung benötigt. Weitere Informationen finden Sie unter [IAM-Rolle für Anwendungen, die auf Amazon EC2-Instances ausgeführt werden](#) im Amazon EC2 Auto Scaling-Benutzerhandbuch.

## Servicebezogene Rollen für AWS die Mainframe-Modernisierung

Unterstützt serviceverknüpfte Rollen	Ja
--------------------------------------	----

Eine serviceverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS-Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Dienstbezogene Rollen werden in Ihrem Dienst angezeigt AWS-Konto und gehören dem Dienst. Ein IAM-Administrator kann die Berechtigungen für Service-verknüpfte Rollen anzeigen, aber nicht bearbeiten.

Einzelheiten zur Erstellung oder Verwaltung servicebezogener Rollen für AWS Mainframe Modernization finden Sie unter [Verwenden von serviceverknüpften Rollen für die Mainframe-Modernisierung](#)

Details zum Erstellen oder Verwalten von serviceverknüpften Rollen finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie in der Tabelle nach einem Service mit einem Yes in der Spalte Service-linked role (Serviceverknüpfte Rolle). Wählen Sie den Link Yes (Ja) aus, um die Dokumentation für die serviceverknüpfte Rolle für diesen Service anzuzeigen.

## Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

Standardmäßig sind Benutzer und Rollen nicht berechtigt, Ressourcen für die AWS Mainframe-Modernisierung zu erstellen oder zu ändern. Sie können auch keine Aufgaben mithilfe der AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) oder ausführen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern die Berechtigung erteilen, Aktionen für die Ressourcen auszuführen, die sie benötigen. Der Administrator kann dann die IAM-Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen annehmen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter [Erstellen von IAM-Richtlinien](#) im IAM-Benutzerhandbuch.

Einzelheiten zu den von AWS Mainframe Modernization definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für jeden Ressourcentyp, finden Sie unter [Aktionen, Ressourcen und Bedingungsschlüssel für die AWS Mainframe-Modernisierung](#) in der Service Authorization Reference.

## Themen

- [Bewährte Methoden für Richtlinien](#)
- [Verwenden der Mainframe Modernization Console AWS](#)
- [Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer](#)

## Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien legen fest, ob jemand AWS Mainframe-Modernisierungsressourcen in Ihrem Konto erstellen, darauf zugreifen oder sie löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Beginnen Sie mit AWS verwalteten Richtlinien und wechseln Sie zu Berechtigungen mit den geringsten Rechten — Verwenden Sie die AWS verwalteten Richtlinien, die Berechtigungen für viele gängige Anwendungsfälle gewähren, um damit zu beginnen, Ihren Benutzern und Workloads Berechtigungen zu gewähren. Sie sind in Ihrem verfügbar. AWS-Konto Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie vom AWS Kunden verwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter [AWS -verwaltete Richtlinien](#) oder [AWS -verwaltete Richtlinien für Auftrags-Funktionen](#) im IAM-Benutzerhandbuch.
- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter [Richtlinien und Berechtigungen in IAM](#) im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Serviceaktionen zu gewähren, wenn diese für einen bestimmten Zweck verwendet werden AWS-Service, z. AWS CloudFormation B. Weitere Informationen finden Sie unter [IAM-JSON-Richtlinienelemente: Bedingung](#) im IAM-Benutzerhandbuch.

- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter [Richtlinienvvalidierung zum IAM Access Analyzer](#) im IAM-Benutzerhandbuch.
- Multi-Faktor-Authentifizierung (MFA) erforderlich — Wenn Sie ein Szenario haben, das IAM-Benutzer oder einen Root-Benutzer in Ihrem System erfordert AWS-Konto, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter [Konfigurieren eines MFA-geschützten API-Zugriffs](#) im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter [Bewährte Methoden für die Sicherheit in IAM](#) im IAM-Benutzerhandbuch.

## Verwenden der Mainframe Modernization Console AWS

Um auf die AWS Mainframe-Modernisierungskonsole zugreifen zu können, benötigen Sie ein Mindestmaß an Berechtigungen. Diese Berechtigungen müssen es Ihnen ermöglichen, Details zu den AWS Mainframe-Modernisierungsressourcen in Ihrem aufzulisten und einzusehen. AWS-Konto Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (Benutzer oder Rollen) mit dieser Richtlinie.

Sie müssen Benutzern, die nur die API AWS CLI oder die AWS API aufrufen, keine Mindestberechtigungen für die Konsole gewähren. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die der API-Operation entsprechen, die die Benutzer ausführen möchten.

Um sicherzustellen, dass Benutzer und Rollen die AWS Mainframe-Modernisierungskonsole weiterhin verwenden können, fügen Sie den Entitäten auch die AWS Mainframe-Modernisierung ConsoleAccess oder die ReadOnly AWS verwaltete Richtlinie hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Berechtigungen zu einem Benutzer](#) im IAM-Benutzerhandbuch.

## Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer

Benutzeridentität angefügt sind. Diese Richtlinie umfasst Berechtigungen zum Ausführen dieser Aktion auf der Konsole oder programmgesteuert mithilfe der OR-API. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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": "*"
    }
  ]
}
```



## Fehlerbehebung bei Identität und Zugriff bei der AWS Mainframe-Modernisierung

Verwenden Sie die folgenden Informationen, um allgemeine Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit AWS Mainframe Modernization und IAM auftreten können.

### Themen

- [Ich bin nicht berechtigt, IAM durchzuführen: PassRole](#)
- [Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Mainframe-Modernisierungsressourcen ermöglichen](#)

### Ich bin nicht berechtigt, IAM durchzuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht autorisiert sind, die `iam:PassRole` Aktion durchzuführen, müssen Ihre Richtlinien aktualisiert werden, damit Sie eine Rolle an AWS Mainframe Modernization übergeben können.

Einige AWS-Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM-Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion im Rahmen der AWS Mainframe-Modernisierung auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

## Ich möchte Personen außerhalb von mir den Zugriff AWS-Konto auf meine AWS Mainframe-Modernisierungsressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Im Fall von Diensten, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (Access Control Lists, ACLs) verwenden, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen darüber, ob die AWS Mainframe-Modernisierung diese Funktionen unterstützt, finden Sie unter [So funktioniert die AWS Mainframe-Modernisierung mit IAM](#)
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen gewähren können, AWS-Konten die Ihnen gehören, finden Sie im IAM-Benutzerhandbuch unter [Gewähren des Zugriffs für einen IAM-Benutzer in einem anderen AWS-Konto , den Sie besitzen](#).
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie [AWS-Konten im IAM-Benutzerhandbuch unter Gewähren des Zugriffs für Dritte](#).
- Informationen dazu, wie Sie über einen Identitätsverbund Zugriff gewähren, finden Sie unter [Gewähren von Zugriff für extern authentifizierte Benutzer \(Identitätsverbund\)](#) im IAM-Benutzerhandbuch.
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontoübergreifenden Zugriff finden Sie unter [So unterscheiden sich IAM-Rollen von ressourcenbasierten Richtlinien](#) im IAM-Benutzerhandbuch.

## Verwenden von serviceverknüpften Rollen für die Mainframe-Modernisierung

AWS Mainframe Modernization [verwendet AWS Identity and Access Management \(IAM\) serviceverknüpfte Rollen](#). Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, die direkt mit der Mainframe-Modernisierung verknüpft ist. Servicebezogene Rollen sind von Mainframe Modernization vordefiniert und beinhalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle erleichtert die Einrichtung der Mainframe-Modernisierung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Die Mainframe-Modernisierung definiert die Berechtigungen ihrer serviceverknüpften Rollen, und sofern nicht anders definiert,

kann nur die Mainframe-Modernisierung ihre Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dadurch werden Ihre Ressourcen für die Mainframe-Modernisierung geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter [AWS -Services, die mit IAM funktionieren](#). Suchen Sie nach den Services, für die Ja in der Spalte Serviceverknüpfte Rollen angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

## Dienstbezogene Rollenberechtigungen für die Mainframe-Modernisierung

Mainframe Modernization verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForAWSM2 — konfigurieren Sie das Netzwerk so, dass es eine Verbindung zu Ihrer VPC herstellt und auf Ressourcen wie Dateisysteme zugreift.

Die serviceverknüpfte Rolle AWSServiceRoleForAWSM 2 vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `m2.amazonaws.com`

Die genannte Rollenberechtigungsrichtlinie AWSM2ServicePolicy ermöglicht es Mainframe Modernization, die folgenden Aktionen an den angegebenen Ressourcen durchzuführen:

- Erstellen, löschen, beschreiben und verknüpfen Sie Berechtigungen für Amazon EC2 EC2-Netzwerkschnittstellen für die Mainframe-Modernisierungsumgebung, um Konnektivität mit der Kunden-VPC herzustellen.
- Registrierung oder Deregistrierung von Einträgen bei Elastic Load Balancing. Auf diese Weise stellen Kunden eine Verbindung zur Mainframe-Modernisierungsumgebung her.
- Beschreiben Sie das Amazon EFS- oder Amazon FSx-Dateisystem, falls verwendet.
- Sendet Metriken CloudWatch aus der Laufzeitumgebung an den Kunden weiter.

```
{
  "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"
      ]
    }
  }
}
```

```
]
}
}
}
]
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine serviceverknüpfte Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter [Serviceverknüpfte Rollenberechtigung](#) im IAM-Benutzerhandbuch.

## Erstellung einer serviceverknüpften Rolle für die Mainframe-Modernisierung

Sie müssen eine serviceverknüpfte Rolle nicht manuell erstellen. Wenn Sie eine Laufzeitumgebung in der AWS Management Console, der oder der AWS API erstellen AWS CLI, erstellt Mainframe Modernization die serviceverknüpfte Rolle für Sie.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen. Wenn Sie eine Laufzeitumgebung erstellen, erstellt Mainframe Modernization die serviceverknüpfte Rolle erneut für Sie.

## Bearbeitung einer serviceverknüpften Rolle für die Mainframe-Modernisierung

Bei der Mainframe-Modernisierung können Sie die `AWSServiceRoleForAWSM` beiden serviceverknüpften Rollen nicht bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollename nach dem Erstellen einer serviceverknüpften Rolle nicht mehr geändert werden. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

## Löschen einer serviceverknüpften Rolle für die Mainframe-Modernisierung

Wenn Sie ein Feature oder einen Dienst, die bzw. der eine serviceverknüpften Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre serviceverknüpften Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

**Note**

Wenn der Mainframe-Modernisierungsdienst die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt das Löschen möglicherweise fehl. Wenn dies passiert, warten Sie einige Minuten und versuchen Sie es erneut.

Um Ressourcen für die Mainframe-Modernisierung zu löschen, die von den beiden verwendet werden `AWSServiceRoleForAWSM`

- Löschen Sie die Laufzeitumgebungen in Mainframe Modernization. Stellen Sie sicher, dass Sie Anwendungen aus einer Umgebung löschen, bevor Sie die Umgebung selbst löschen.

So löschen Sie die serviceverknüpfte Rolle mit IAM

Verwenden Sie die IAM-Konsole, die oder die AWS API AWS CLI, um die `AWSServiceRoleForAWSM` beiden dienstverknüpften Rollen zu löschen. Weitere Informationen finden Sie unter [Löschen einer serviceverknüpften Rolle](#) im IAM-Benutzerhandbuch.

Unterstützte Regionen für serviceverknüpfte Mainframe-Modernisierungsrollen

Die Mainframe-Modernisierung unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter [AWS -Regionen und -Endpunkte](#).

## Überprüfung der Einhaltung der Vorschriften für AWS die Mainframe-Modernisierung

Externe Prüfer bewerten die Sicherheit und Konformität der AWS Mainframe-Modernisierung im Rahmen mehrerer AWS Compliance-Programme. Hierzu zählen unter anderem SOC, PCI, FedRAMP und HIPAA.

Eine Liste der AWS Services im Rahmen bestimmter Compliance-Programme finden Sie unter [AWS-Services in Umfang nach Compliance-Programm](#) . Allgemeine Informationen finden Sie unter [AWS - Compliance-Programme](#).

Sie können Prüfberichte von Drittanbietern unter heruntergeladen AWS Artifact. Weitere Informationen finden Sie unter [Berichte heruntergeladen unter](#) .

Ihre Verantwortung für die Einhaltung von Vorschriften bei der Nutzung von AWS Mainframe Modernization hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung von Vorschriften unterstützen:

- [Schnellstartanleitungen für Sicherheit und Compliance](#) – In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von sicherheits- und konformitätsorientierten Basisumgebungen auf AWS angegeben.
- Whitepaper „[Architecting for HIPAA Security and Compliance](#)“ — In diesem Whitepaper wird beschrieben, wie Unternehmen HIPAA-konforme Anwendungen erstellen können AWS .
- [AWS Compliance-Ressourcen](#) — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- [Bewertung von Ressourcen anhand von Regeln](#) im AWS Config Entwicklerhandbuch — AWS Config; bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- [AWS Security Hub](#)— Dieser AWS Service bietet einen umfassenden Überblick über Ihren Sicherheitsstatus und hilft Ihnen AWS , die Einhaltung der Sicherheitsstandards und Best Practices der Branche zu überprüfen.

## Resilienz bei der AWS Mainframe-Modernisierung

Die AWS globale Infrastruktur basiert auf AWS Regionen und Availability Zones. Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen zu AWS Regionen und Availability Zones finden Sie unter [AWS Globale Infrastruktur](#).

## Sicherheit der Infrastruktur in AWS Mainframe Modernization

Als verwalteter Service AWS Mainframe Modernization ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur

finden Sie unter [AWS Cloud-Sicherheit](#). Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API-Aufrufe, um über das Netzwerk auf Mainframe Modernization zuzugreifen. Kunden müssen Folgendes unterstützen:

- Transport Layer Security (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Verschlüsselungs-Suiten mit Perfect Forward Secrecy (PFS) wie DHE (Ephemeral Diffie-Hellman) oder ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Außerdem müssen Anforderungen mit einer Zugriffsschlüssel-ID und einem geheimen Zugriffsschlüssel signiert sein, der einem IAM-Prinzipal zugeordnet ist. Alternativ können Sie mit [AWS Security Token Service](#) (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

## Zugriff AWS Mainframe Modernization über einen Schnittstellenendpunkt ( )AWS PrivateLink

Sie können AWS PrivateLink damit eine private Verbindung zwischen Ihrer VPC und AWS Mainframe Modernization herstellen. Sie können auf Mainframe Modernization wie in Ihrer VPC zugreifen, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung verwenden zu müssen. Instances in Ihrer VPC benötigen keine öffentlichen IP-Adressen, um auf Mainframe Modernization zuzugreifen.

Sie stellen diese private Verbindung her, indem Sie einen Schnittstellen-Endpunkt erstellen, der von AWS PrivateLink unterstützt wird. Wir erstellen eine Endpunkt-Netzwerkschnittstelle in jedem Subnetz, das Sie für den Schnittstellen-Endpunkt aktivieren. Dabei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Einstiegspunkt für den Datenverkehr dienen, der für die Mainframe-Modernisierung bestimmt ist.

Weitere Informationen finden Sie im Handbuch unter [Access AWS-Services through AWS PrivateLink](#).AWS PrivateLink



## Überlegungen zur Mainframe-Modernisierung

Bevor Sie einen Schnittstellen-Endpunkt für die Mainframe-Modernisierung einrichten, lesen Sie die [Überlegungen](#) im AWS PrivateLink Handbuch.

Mainframe Modernization unterstützt Aufrufe aller API-Aktionen über den Schnittstellenendpunkt.

## Erstellen Sie einen Schnittstellenendpunkt für die Mainframe-Modernisierung

Sie können einen Schnittstellenendpunkt für die Mainframe-Modernisierung entweder mit der Amazon VPC-Konsole oder mit AWS Command Line Interface (AWS CLI) erstellen. Weitere Informationen finden Sie unter [Erstellen eines Schnittstellenendpunkts](#) im AWS PrivateLink -Leitfaden.

Erstellen Sie einen Schnittstellenendpunkt für die Mainframe-Modernisierung mit dem folgenden Servicenamen:

```
com.amazonaws.region.m2
```

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an Mainframe Modernization stellen, indem Sie den standardmäßigen regionalen DNS-Namen verwenden. z. B. `m2.us-east-1.amazonaws.com`.

## Erstellen einer Endpunktrichtlinie für Ihren Schnittstellen-Endpunkt

Eine Endpunktrichtlinie ist eine IAM-Ressource, die Sie an einen Schnittstellen-Endpunkt anfügen können. Die standardmäßige Endpunktrichtlinie ermöglicht den vollen Zugriff auf Mainframe Modernization über den Schnittstellenendpunkt. Um den Zugriff auf die Mainframe-Modernisierung von Ihrer VPC aus zu kontrollieren, fügen Sie dem Schnittstellenendpunkt eine benutzerdefinierte Endpunktrichtlinie hinzu.

Eine Endpunktrichtlinie gibt die folgenden Informationen an:

- Die Principals, die Aktionen ausführen können (AWS-Konten, Benutzer und IAM-Rollen).
- Aktionen, die ausgeführt werden können
- Die Ressourcen, auf denen die Aktionen ausgeführt werden können.

Weitere Informationen finden Sie unter [Steuern des Zugriffs auf Services mit Endpunktrichtlinien](#) im AWS PrivateLink -Leitfaden.

## Beispiel: VPC-Endpunktrichtlinie für Mainframe-Modernisierungsaktionen

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellenendpunkt anhängen, gewährt sie allen Prinzipalen auf allen Ressourcen Zugriff auf die aufgelisteten Mainframe-Modernisierungsaktionen.

```
//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": "*"
  }
]
```

# Überwachung der AWS Mainframe-Modernisierung

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung von AWS Mainframe Modernization und Ihren anderen AWS-Lösungen. AWS bietet die folgenden Überwachungstools, um die AWS Mainframe-Modernisierung zu beobachten, zu melden, wenn etwas nicht stimmt, und gegebenenfalls automatische Maßnahmen zu ergreifen:

- Amazon CloudWatch überwacht Ihre AWS Ressourcen und die Anwendungen, auf denen Sie laufen, AWS in Echtzeit. Sie können Kennzahlen erfassen und verfolgen, benutzerdefinierte Dashboards erstellen und Alarmer festlegen, die Sie benachrichtigen oder Maßnahmen ergreifen, wenn eine bestimmte Metrik einen von Ihnen festgelegten Schwellenwert erreicht. Sie können beispielsweise die CPU-Auslastung oder andere Kennzahlen Ihrer Amazon EC2 EC2-Instances CloudWatch verfolgen und bei Bedarf automatisch neue Instances starten. Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).
- Mit Amazon CloudWatch Logs können Sie Ihre Protokolldateien von Amazon EC2 EC2-Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs können Informationen in den Protokolldateien überwachen und Sie benachrichtigen, wenn bestimmte Schwellenwerte erreicht werden. Sie können Ihre Protokolldaten auch in einem sehr robusten Speicher archivieren. Weitere Informationen finden Sie im [Amazon CloudWatch Logs-Benutzerhandbuch](#).
- AWS CloudTrailerfasst API-Aufrufe und zugehörige Ereignisse, die von oder im Namen Ihres AWS Kontos getätigt wurden, und übermittelt die Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket. Sie können feststellen, welche Benutzer und Konten angerufen wurden AWS, von welcher Quell-IP-Adresse aus die Anrufe getätigt wurden und wann die Aufrufe erfolgten. Weitere Informationen finden Sie im [AWS CloudTrail -Benutzerhandbuch](#).

## Überwachung der AWS Mainframe-Modernisierung mit Amazon CloudWatch

Sie können die AWS Mainframe-Modernisierung mithilfe von CloudWatch Tools überwachen. Dabei werden Rohdaten gesammelt und zu lesbaren Metriken verarbeitet, die nahezu in Echtzeit verfügbar sind. Diese Statistiken werden 15 Monate gespeichert, damit Sie auf Verlaufsdaten zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarmer einrichten, die auf bestimmte Grenzwerte achten und

Benachrichtigungen senden oder Aktivitäten auslösen, wenn diese Grenzwerte erreicht werden.

Weitere Informationen finden Sie im [CloudWatch Amazon-Benutzerhandbuch](#).

In den folgenden Tabellen sind die Metriken und Dimensionen für die AWS Mainframe-Modernisierung aufgeführt. Der Namespace für diese Metriken lautet. AWS/M2

## Metriken zur Laufzeitumgebung

Metrik	Beschreibung
CPUUtilization	<p>Die CPU-Auslastung von Instanzen in der Umgebung.</p> <p>Dimension: EnvironmentID</p> <p>Einheiten: Prozent</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>
InboundNetworkThroughput	<p>Eingehender Netzwerkdurchsatz von Instanzen in der Umgebung.</p> <p>Dimension: EnvironmentID</p> <p>Einheiten: Byte pro Sekunde</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>
MemoryUtilization	<p>Die Speicherauslastung von Instanzen in der Umgebung.</p> <p>Dimension: EnvironmentID</p> <p>Einheiten: Prozent</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>

Metrik	Beschreibung
OutboundNetworkThroughput	<p>Ausgehender Netzwerkdurchsatz der Instanzen in der Umgebung.</p> <p>Dimension: EnvironmentID</p> <p>Einheiten: Byte pro Sekunde</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>

## Anwendungsmetriken

Metrik	Beschreibung
BatchJobCompletedCount	<p>Die Anzahl der abgeschlossenen Jobs während des Zeitintervalls.</p> <p>Diese Metrik ist für Micro Focus und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
BatchJobFailedCount	<p>Die Anzahl der fehlgeschlagenen Jobs während des Zeitintervalls.</p> <p>Diese Metrik ist für Micro Focus und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>

Metrik	Beschreibung
JvmMemoryFree	<p>Die Menge des verfügbaren Speichers, der derzeit nicht von der Java Virtual Machine verwendet wird.</p> <p>Diese Metrik ist nur für die AWS Blu Age Runtime Engine verfügbar. Sie ist für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>
JvmMemoryMax	<p>Die maximal zulässige Speichermenge für die Java Virtual Machine.</p> <p>Diese Metrik ist nur für die AWS Blu Age Runtime Engine verfügbar. Sie ist für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>

Metrik	Beschreibung
JvmMemoryUsed	<p>Die Menge an Arbeitsspeicher, die von der Java Virtual Machine aktiv genutzt wird.</p> <p>Diese Metrik ist nur für die AWS Blu Age Runtime Engine verfügbar. Sie ist für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Byte</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>
ProcessesActiveCount	<p>Die aktive Anzahl gleichzeitiger Dienstausführungsprozesse, die Anfragen verarbeiten.</p> <p>Diese Metrik ist nur für die Micro Focus Runtime Engine verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
SessionCount	<p>Die Anzahl der HTTP-Sitzungen für die Anwendung.</p> <p>Diese Metrik ist nur für die AWS Blu Age Runtime Engine verfügbar. Sie ist für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>

Metrik	Beschreibung
SharedMemoryFree	<p>Der Speicher, der dem Unternehmensserver zur Verfügung steht, um alle Informationen zu speichern, die er zur Ausführung von Transaktionen und Aufträgen benötigt.</p> <p>Diese Metrik ist nur für die Micro Focus Runtime Engine verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>
ThreadActiveCount	<p>Die Anzahl der Engine-Threads, die Anfragen verarbeiten.</p> <p>Diese Metrik ist nur für die AWS Blu Age Runtime Engine verfügbar. Sie ist für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>



Metrik	Beschreibung
TransactionCompletedCount	<p>Die Anzahl der festgeschriebenen Transaktionen während des Zeitintervalls.</p> <p>Diese Metrik ist für Micro Focus und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
TransactionFailedCount	<p>Die Anzahl der fehlgeschlagenen Transaktionen während des Zeitintervalls.</p> <p>Diese Metrik ist für Micro Focus und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Anzahl</p> <p>Gültige Statistiken: Summe</p>
TransactionResponseTime	<p>Die Zeitspanne zwischen dem Zeitpunkt, an dem ein Benutzer eine Anfrage sendet, bis zu dem Zeitpunkt, zu dem die Anwendung angibt, dass die Anfrage abgeschlossen wurde.</p> <p>Diese Metrik ist für Micro Focus und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.</p> <p>Dimension: applicationId</p> <p>Einheiten: Millisekunden</p> <p>Gültige Statistiken: Durchschnitt, Minimum, Maximum</p>

## Dimensionen

Dimension	Beschreibung
applicationId	Diese Dimension filtert die Metrik anhand der ID auf die identifizierte Anwendung.
Umgebungs-ID	Diese Dimension filtert die Metrik anhand der ID auf die identifizierte Umgebung.

## Protokollierung von API-Aufrufen für die AWS Mainframe-Modernisierung mit AWS CloudTrail

AWS Mainframe Modernization ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Aktionen bereitstellt, die von einem Benutzer, einer Rolle oder einem AWS Service im Rahmen der AWS Mainframe-Modernisierung ausgeführt wurden. CloudTrail erfasst alle API-Aufrufe für die AWS Mainframe-Modernisierung als Ereignisse. Zu den erfassten Aufrufen gehören Aufrufe von der AWS Mainframe-Modernisierungskonsole und Codeaufrufen für die AWS Mainframe-Modernisierungs-API-Operationen. Wenn Sie einen Trail erstellen, können Sie die kontinuierliche Bereitstellung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren, einschließlich Ereignissen für die AWS Mainframe-Modernisierung. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse trotzdem in der CloudTrail Konsole im Ereignisverlauf anzeigen. Anhand der von CloudTrail gesammelten Informationen können Sie die Anfrage an die AWS Mainframe-Modernisierung, die IP-Adresse, von der aus die Anfrage gestellt wurde, wer die Anfrage gestellt hat, wann sie gestellt wurde, und weitere Details ermitteln.

Weitere Informationen CloudTrail dazu finden Sie im [AWS CloudTrail Benutzerhandbuch](#).

## AWS Informationen zur Mainframe-Modernisierung finden Sie unter CloudTrail

CloudTrail ist für Ihr AWS Konto aktiviert, wenn Sie das Konto erstellen. Wenn bei der AWS Mainframe-Modernisierung eine Aktivität auftritt, wird diese Aktivität zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie in einem Ereignis aufgezeichnet. Sie können aktuelle Ereignisse in Ihrem AWS Konto ansehen, suchen und herunterladen. Weitere Informationen finden Sie unter [Ereignisse mit CloudTrail Ereignisverlauf anzeigen](#).

Für eine fortlaufende Aufzeichnung von Ereignissen in Ihrem AWS Konto, einschließlich Ereignissen im Zusammenhang mit der AWS Mainframe-Modernisierung, erstellen Sie einen Trail. Ein Trail ermöglicht CloudTrail die Übermittlung von Protokolldateien an einen Amazon S3 S3-Bucket. Wenn Sie einen Pfad in der Konsole anlegen, gilt dieser für alle AWS-Regionen. Der Trail protokolliert Ereignisse aus allen Regionen der AWS Partition und übermittelt die Protokolldateien an den von Ihnen angegebenen Amazon S3 S3-Bucket. Darüber hinaus können Sie andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie hier:

- [Übersicht zum Erstellen eines Trails](#)
- [CloudTrail unterstützte Dienste und Integrationen](#)
- [Konfiguration von Amazon SNS SNS-Benachrichtigungen für CloudTrail](#)
- [Empfangen von CloudTrail Protokolldateien aus mehreren Regionen](#)
- [Empfangen von CloudTrail Protokolldateien von mehreren Konten](#)

Alle AWS Mainframe-Modernisierungsaktionen werden von der Mainframe [Modernization API Reference](#) protokolliert CloudTrail und sind in AWS dieser Dokumentation dokumentiert. Beispielsweise generieren Aufrufe von `CreateEnvironment` und `CreateDeployment` Aktionen Einträge in den CloudTrail Protokolldateien. `CreateApplication`

Jeder Ereignis- oder Protokolleintrag enthält Informationen zu dem Benutzer, der die Anforderung generiert hat. Die Identitätsinformationen unterstützen Sie bei der Ermittlung der folgenden Punkte:

- Ob die Anfrage mit Anmeldeinformationen des Root-Benutzers oder des Benutzers gestellt wurde.
- Gibt an, ob die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen Verbundbenutzer gesendet wurde.
- Ob die Anfrage von einem anderen AWS Dienst gestellt wurde.

Weitere Informationen finden Sie unter dem [CloudTrail UserIdentity-Element](#).

## Grundlegendes zu Einträgen in der Protokolldatei zur AWS Mainframe-Modernisierung

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an einen von Ihnen angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen

oder mehrere Protokolleinträge. Ein Ereignis stellt eine einzelne Anforderung aus einer beliebigen Quelle dar und enthält Informationen über die angeforderte Aktion, Datum und Uhrzeit der Aktion, Anforderungsparameter usw. CloudTrail Protokolldateien sind kein geordneter Stack-Trace der öffentlichen API-Aufrufe, sodass sie nicht in einer bestimmten Reihenfolge angezeigt werden.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag, der die `CreateApplication` Aktion demonstriert.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAI6WZTHGYAEXAMPLE",
    "arn": "arn:aws:sts::444455556666:assumed-role/Admin/Mary_Major",
    "accountId": "444455556666",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAI6WZTHGYAEXAMPLE",
        "arn": "arn:aws:iam::444455556666:role/Admin",
        "accountId": "444455556666",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2022-06-01T20:38:22Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2022-06-01T20:40:39Z",
  "eventSource": "m2.amazonaws.com",
  "eventName": "CreateApplication",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.196.65",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101 Firefox/91.0",
  "requestParameters": {
    "clientToken": "1abc23de-f45g-6789-h01i-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/lsfhw7fffrosff2lncwqcu",
    "applicationId": "lsfhw7fffrosff2lncwqcu"
  },
  "requestID": "36982d38-fcde-4bfe-a89a-7bd78d43c926",
  "eventID": "d7f0fc36-46ae-4157-9a79-c79f385fda98",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "444455556666",
  "eventCategory": "Management"
}
```

# Fehlerbehebung

Mithilfe der Informationen in diesem Abschnitt können Sie häufig auftretende Fehler in Anwendungen und Laufzeitumgebungen zur AWS Mainframe-Modernisierung mithilfe der AWS Blu Age- und Micro Focus-Engines beheben.

## Themen

- [Fehler: Beim Warten auf die Entsperrung des Datensatznamens ist eine Zeitüberschreitung aufgetreten](#)
- [Auf die URL einer Anwendung kann nicht zugegriffen werden](#)
- [AWS Blu Insights wird nicht von der Konsole aus geöffnet](#)

## Fehler: Beim Warten auf die Entsperrung des Datensatznamens ist eine Zeitüberschreitung aufgetreten

- Motor: AWS Blu Age
- Komponente: Blusam

Wenn Sie diesen Fehler in den CloudWatch Amazon-Protokollen für eine AWS Mainframe-Modernisierungsanwendung sehen, die die AWS Blu Age-Engine verwendet und in einer Umgebung mit dem Hochverfügbarkeitsmuster ausgeführt wird, deutet dies darauf hin, dass eine andere Anwendung einen gemeinsam genutzten Datensatz gesperrt hat. In der Regel tritt diese Situation auf, wenn die andere Anwendung abstürzt oder auf andere Weise ausfällt und die Sperre nicht aufgehoben wird.

## Wie tritt dieser Fehler auf

Die Anwendung `example-app-1` versucht, einen Datensatz `example-record-1` für einen Schreibvorgang zu sperren. Durch diesen Vorgang wird sowohl eine Sperre für die Datenmenge `example-dataset-1`, die Eigentümer ist `example-record-1`, als auch eine Sperre für sich `example-record-1` selbst erstellt. Jetzt versucht eine andere Anwendung `example-app-2`, denselben Datensatz zu sperren `example-record-1`. Die Datenmenge und der Datensatz sind bereits gesperrt. `example-app-2` Warten Sie also, bis die Sperre aufgehoben wird. Bei einem `example-app-1` Absturz besteht die Sperre für die Datenmenge `example-dataset-1`

immer noch, was `example-app-2` dazu führt, dass der Schreibversuch abgebrochen und eine Timeout-Ausnahme ausgelöst wird. Diese Deadlock-Situation verhindert, dass alle Anwendungen die Verbindung erreichen. `example-dataset-1`

## Woher wissen Sie, ob dies Ihre Situation ist?

Suchen Sie nach einer fehlgeschlagenen Anwendung und überprüfen Sie, ob sie denselben Datensatz verwendet, der in der Fehlermeldung erwähnt wurde. Prüfen Sie, ob die Anwendung in einer Laufzeitumgebung mit dem Hochverfügbarkeitsmuster ausgeführt wird. Die Anwendung, die die Timeout-Ausnahme ausgelöst hat, kann nicht fortfahren und zeigt den `Failed` Status an.

## Was können Sie tun?

Um die Situation sofort zu lösen, können Sie das Öffnen des Schlosses erzwingen. Um zu verhindern, dass in future eine ähnliche Situation auftritt, können Sie zwei Parameter konfigurieren, die den auto Reparaturmechanismus von Blusam steuern.

## Erzwingen Sie das Lösen des Schlosses

Der Blusam Lock Manager verwendet Amazon ElastiCache for Redis, um gemeinsam genutzte Sperren zwischen Anwendungen bereitzustellen. Verwenden Sie das Redis-CLI-Hilfsprogramm ElastiCache, um Sperren aufzuheben. Sie können eine einzelne Datensatzsperre nicht löschen. Sie müssen alle Sperren aus der Datenmenge entfernen, die der Eigentümer ist. Führen Sie folgende Schritte aus:

1. Stellen Sie ElastiCache mit dem folgenden Befehl eine Connect zu Ihrem her:

```
redis-cli -h hostname -p port
```

Sie finden die Details Ihres ElastiCache in der ElastiCache Konsole unter <https://console.aws.amazon.com/elasticache/>.

2. Geben Sie Ihr Passwort ein.
3. Geben Sie den Befehl, den Sie ausführen möchten, wie folgt ein:

Befehl	Zweck
KEYS *	Ruft alle vorhandenen Schlüssel ab.

Befehl	Zweck
SCHLÜSSEL * <i>YOUR_DATASET_NAME</i>	Holen Sie sich einen Datensatz-Sperrschlüssel.
LÖSCHEN SIE <i>THE_RETURNED_KEY</i>	Löscht eine Datensatzsperrschlüssel.
FLUSHDB	Säubere das gesamte Redis.  <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9f9;"> <p><b>⚠ Warning</b></p> <p>Alle Daten im Redis-Cache gehen verloren. Wenn Redis für andere Zwecke verwendet wird, z. B. für die Verarbeitung von HTTP-Sitzungen, möchten Sie es möglicherweise nicht verwenden. FLUSHDB</p> </div>

## Konfigurieren Sie den auto Reparaturmechanismus von Blusam

Der Blusam Locks Manager beinhaltet einen auto Reparaturmechanismus, um Deadlocks bei Datensätzen oder Datensätzen zu verhindern. Sie können die folgenden Parameter in der Anwendungsdefinition (`application-main.yml`) anpassen, um den auto Reparaturmechanismus zu konfigurieren:

- `locksDeadTime`: bezieht sich auf die maximale Zeit, für die eine Anwendung eine Sperre aufrechterhalten kann. Nach Ablauf dieser Zeit wird die Sperre für abgelaufen erklärt und sofort aufgehoben. Der `locksDeadTime` Wert wird in Millisekunden angegeben und der Standardwert ist 1000.
- `locksCheck`: definiert die Blusam Locks Manager-Strategie zur Überprüfung von Sperren. Alle Blusam-Locks ElastiCache sind mit einem Zeitstempel versehen und haben eine Ablaufzeit. Der `locksCheck` Parameterwert bestimmt, ob abgelaufene Sperren entfernt werden.
  - `off`: Es wird zu keinem Zeitpunkt eine Prüfung ausgeführt. Es können Deadlocks auftreten. (Nicht empfohlen)
  - `reboot`: Prüfungen werden ausgeführt, wenn eine AWS Mainframe-Modernisierungs-Anwendungsinstanz, die in einer AWS Mainframe-Modernisierungs-Laufzeitumgebung



ausgeführt wird, gestartet oder neu gestartet wird. Alle abgelaufenen Sperren werden sofort aufgehoben. (Standard)

- **timeout:** Prüfungen werden ausgeführt, wenn eine AWS Mainframe-Modernisierungs-Anwendungsinstanz, die in einer AWS Mainframe-Modernisierungs-Laufzeitumgebung ausgeführt wird, gestartet oder neu gestartet wird oder wenn beim Versuch, einen Datensatz zu sperren, ein Timeout abläuft. Abgelaufene Sperren werden sofort freigegeben.

Weitere Informationen zur Anwendungsdefinition für eine AWS Blu Age-Anwendung finden Sie unter [AWS Beispiel für eine Blu-Age-Anwendungsdefinition](#).

## Blusam Locks Manager

Im Kontext einer Laufzeitumgebung für die AWS Mainframe-Modernisierung, die das Hochverfügbarkeitsmuster verwendet, kann eine AWS Blu Age-Anwendung mehrfach bereitgestellt werden. Bei Anwendungen, die Blusam-Datensätze verarbeiten, können Probleme beim gleichzeitigen Zugriff auftreten. Der Blusam Locks Manager gewährleistet die Datenintegrität und verwaltet den Lese- und Schreibzugriff auf Datensätze und Datensätze, indem er gemeinsame Sperren zwischen Anwendungen bereitstellt, die verwenden. ElastiCache Dieser Mechanismus ermöglicht es mehr als einer Anwendung, den Datensatz gleichzeitig zu lesen, und stellt sicher, dass jeweils nur eine Anwendung den Datensatz schreibt.

### Sperren schreiben

Um einen bestimmten Datensatz zu aktualisieren oder zu löschen, muss die Anwendung zuerst die Datenmenge sperren, der der Datensatz gehört, und dann den Datensatz selbst sperren. Wenn der Datensatz gesperrt ist, wird die Datenmengensperre aufgehoben, und andere Datensätze aus derselben Datenmenge können verwendet werden. Wenn der Aktualisierungs- oder Löschvorgang abgeschlossen ist, wird die Sperre für den Datensatz aufgehoben. Es kann jeweils nur eine Anwendung den Datensatz aktualisieren, wodurch andere Anwendungen entweder am Lesen oder Schreiben gehindert werden, bis die Sperre aufgehoben wird, sofern die definierte Anwendungsrichtlinie das Warten auf die Freigabe zulässt.

### Sperren lesen

Solange für den Datensatz oder die Datenmenge keine Schreibsperre gilt, können mehrere Anwendungen dieselben Datensätze gleichzeitig lesen. Um einen Datensatz für einen Schreibvorgang zu sperren, müssen alle Lesesperren aufgehoben werden.

**Note**

Der Blusam Locks Manager verwaltet den Zugriff von mehreren Threads in einer bestimmten Anwendung mithilfe desselben Sperrmechanismus.

## Auf die URL einer Anwendung kann nicht zugegriffen werden

- Motor: AWS Blu Age und Micro Focus
- Komponente: Anwendungen

Wenn Sie nicht auf die URL einer laufenden AWS Mainframe-Modernisierungsanwendung zugreifen können, die Sie erstellt und in einer AWS Mainframe-Modernisierungs-Laufzeitumgebung bereitgestellt haben, müssen Sie möglicherweise die Regeln für eingehende Nachrichten in der Sicherheitsgruppe konfigurieren, die Sie der Laufzeitumgebung zugeordnet haben.

### Wie tritt dieser Fehler auf

Wenn Sie eine Laufzeitumgebung erstellen, müssen für die von Ihnen bereitgestellte Sicherheitsgruppe, einschließlich der Standardsicherheitsgruppe, eingehende Regeln konfiguriert sein, um Datenverkehr zu den bereitgestellten Anwendungen von außerhalb der VPC zuzulassen, wenn Sie diese Art von Zugriff zulassen möchten.

### Woher wissen Sie, ob dies Ihre Situation ist?

Die Anwendung wurde erfolgreich gestartet und läuft normal, aber Sie können über ihre URL keine Verbindung zu ihr herstellen.

### Was können Sie tun?

Prüfen Sie, ob die Amazon VPC-Sicherheitsgruppe, die der Laufzeitumgebung zugeordnet ist, Datenverkehr zur Umgebung über die entsprechenden Anwendungsporen zulässt. Gehen Sie wie folgt vor, um die Sicherheitsgruppenregeln zu überprüfen:

1. Öffnen Sie die AWS Mainframe Modernization Console unter <https://console.aws.amazon.com/m2/>.
2. Wählen Sie in der linken Navigationsleiste Umgebungen aus.

3. Wählen Sie die Laufzeitumgebung aus, die die Anwendung hostet, zu der Sie eine Verbindung herstellen möchten.
4. Wählen Sie Konfigurationen.
5. Wählen Sie unter Sicherheit und Netzwerk die Sicherheitsgruppe aus. Der Link öffnet die Details der Sicherheitsgruppe in der Amazon VPC-Konsole.
6. Wählen Sie bei Bedarf „Regeln für eingehenden Datenverkehr bearbeiten“ und fügen Sie die folgende Regel hinzu, falls sie noch nicht vorhanden ist:

#### Typ

Custom TCP

#### Port

8196 oder der Port, der den in der Anwendungsdefinition angegebenen Listener-Eigenschaften entspricht. Weitere Informationen finden Sie unter [Schritt 2: Erstellen Sie die Anwendungsdefinition](#).

#### Quelle

Die IP-Adresse, von der aus Sie die Anwendung aufrufen. Sie können MyIP aus dem Drop-down-Menü auswählen. Wenn Sie immer noch Probleme mit dem Timeout haben, versuchen Sie, Anywhere IPV4 oder Anywhere IPV6 auszuwählen. Stellen Sie sicher, dass Sie die Anwendung beenden und erneut starten, nachdem Sie die Regel für eingehenden Datenverkehr zur Sicherheitsgruppe hinzugefügt haben.

Weitere Informationen finden Sie unter [Arbeiten mit Sicherheitsgruppenregeln](#) im Amazon VPC-Benutzerhandbuch.

## AWS Blu Insights wird nicht von der Konsole aus geöffnet

- Motor: AWS Blu Age
- Komponente: Blu Insights

Wenn Sie versuchen, über die AWS Mainframe Modernization Console auf Blu Insights zuzugreifen, wird diese nicht geöffnet und der neue Tab wird sofort geschlossen.

## Wie tritt dieser Fehler auf

Die Rolle, die Sie für den Zugriff auf Blu Insights verwenden, verfügt nicht über ausreichende Berechtigungen.

## Was können Sie tun?

Fügen Sie der Rolle eine IAM-Richtlinie hinzu, damit sie auf Blu Insights zugreifen kann. Stellen Sie sicher, dass die Richtlinie mindestens die folgenden Berechtigungen umfasst.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "m2:GetSignedBluinsightsUrl"
      ],
      "Resource": "arn:aws:m2:region:account:*"
    }
  ]
}
```

Stellen Sie sicher, dass Sie `region` und `account` durch das richtige AWS-Region und ersetzen AWS-Konto.

# Dokumentverlauf für das AWS Mainframe Modernization User Guide

In der folgenden Tabelle werden die Dokumentationsversionen für AWS Mainframe Modernization beschrieben.

Änderung	Beschreibung	Datum
<a href="#">Das Tutorial zu Managed Runtime für Micro Focus wurde aktualisiert</a>	Dieses Tutorial zeigt, wie Sie die CardDemo Beispielanwendung in einer von AWS Mainframe Modernization verwalteten Laufzeitumgebung mit der Micro Focus Runtime Engine bereitstellen und ausführen.	5. Februar 2024
<a href="#">Versionshinweise für AWS Blue Age Runtime and Modernization Tools Version 3.9.0.</a>	Diese Version von AWS Blue Age Runtime and Modernization Tools konzentriert sich auf mehrere bereichsübergreifende Verbesserungen im gesamten Produkt, mit denen die Leistung in Hochverfügbarkeitsarchitekturen gesteigert werden soll, sowie auf neue Funktionen, um die Ausführung von Jobs auf die nächste Stufe zu heben.	18. Dezember 2023
<a href="#">Übertragen Sie Dateien zwischen Mainframe und AWS</a>	Neue Funktion zum Übertragen von Dateien vom Quell-Mainframe auf AWS	8. November 2023
<a href="#">Transaktionen für Anwendungen verwalten</a>	Neue Funktion zum Anzeigen und Bearbeiten von Transaktionen	16. Oktober 2023

onen für Anwendungen zur AWS Mainframe-Modernisierung veröffentlicht.

[Versionshinweise für AWS Blu Age Runtime and Modernization Tools Version 3.6.0.](#)

Diese Version von AWS Blu Age Runtime and Modernization Tools bietet neue Funktionen für ältere Migrationen von zOS und AS400. Diese sind hauptsächlich darauf ausgerichtet, die CICS-Unterstützungsmechanismen zu erweitern, die JCL-Funktionen zu ergänzen, die Leistung bei gleichzeitigen und umfangreichen Funktionen zu optimieren und Funktionen hinzuzufügen. multi-data-source

4. August 2023

[Sie können jetzt eine neue Version einer Anwendung bereitstellen, wenn die Anwendung beendet ist.](#)

Bisher mussten Sie zum Bereitstellen einer neuen Version einer Anwendung die bereitgestellte Version löschen. Jetzt können Sie die bereitgestellte Version einfach beenden und eine neue Version bereitstellen.

26. Juli 2023

[AWS Blu Age Runtime-Paket für eine einfachere Amazon EC2 EC2-Bereitstellung](#)

AWS Mainframe-Modernisierung mit AWS Blu Age Runtime ist jetzt mit mehr Flexibilität für die Konfiguration des gesamten Stacks und die Bereitstellung auf Amazon EC2 EC2-Instances in Ihrem verfügbar. AWS-Konto

6. Juli 2023

---

<a href="#">Einmaliges Anmelden bei AWS AWS Blu Age Blu Insights.</a>	AWS AWS Blu Age Blu Insights ist ab dem Single AWS Management Console Sign-On verfügbar.	31. März 2023
<a href="#">GA-Veröffentlichung</a>	GA-Version des AWS Mainframe Modernization User Guide.	8. Juni 2022
<a href="#">Erstversion</a>	Erste Version (öffentliche Vorschau) des AWS Mainframe Modernization User Guide.	30. November 2021

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.