Benutzerhandbuch

# AWS Mainframe-Modernisierung



Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Mainframe-Modernisierung: Benutzerhandbuch

Copyright © 2025 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 die AWS Mainframe-Modernisierung ..... 5
Verwenden Sie Mainframe Modernization zum ersten Mal? AWS ..... 5
Preise für AWS die Mainframe-Modernisierung ..... 5
Für die AWS Mainframe-Modernisierung gerüstet ..... 6
Melden Sie sich für ein AWS-Konto ..... 6
Erstellen eines Benutzers mit Administratorzugriff ..... 7
Konzepte ..... 9
Anwendung ..... 9
Definition der Anwendung ..... 10
Batch-Auftrag ..... 10
Konfiguration ..... 11
Dataset ..... 11
Umgebung ..... 11
Mainframe-Modernisierung ..... 11
Reise zur Migration ..... 11
Montagepunkt ..... 12
Automatisiertes Refactoring ..... 12
Plattformwechsel ..... 12
Ressource ..... 12
Runtime-Engine ..... 12
Modernisierungsansatz ..... 13
Phase bewerten ..... 13
Phase der Mobilisierung ..... 14
Phase der Migration und Modernisierung ..... 14
Phase „Betrieb und Optimierung“ ..... 15
Erste Schritte ..... 16
Tutorial: Verwaltete Laufzeit für AWS Blu Age einrichten ..... 16
Voraussetzungen ..... 17
Schritt 1: Laden Sie die Demo-Anwendung hoch ..... 17

Schritt 2: Erstellen Sie die Anwendungsdefinition ... 17
Schritt 3: Erstellen Sie eine Laufzeitumgebung ... 18
Schritt 4: Erstellen Sie eine Anwendung ... 23
Schritt 5: Stellen Sie eine Anwendung bereit ... 25
Schritt 6: Starten Sie eine Anwendung ... 27
Schritt 7: Greifen Sie auf die Anwendung zu ... 27
Schritt 8: Testen der Anwendung ... 28
Bereinigen von -Ressourcen ... 30
Tutorial: Richten Sie die verwaltete Laufzeit für Rocket Software ein ... 30
Voraussetzungen ... 31
Schritt 1: Einen Amazon S3 S3-Bucket erstellen und laden ... 32
Schritt 2: Eine Datenbank erstellen und konfigurieren ... 33
Schritt 3: Erstellen und konfigurieren Sie ein AWS KMS key ... 35
Schritt 4: Erstellen und konfigurieren Sie einen geheimen AWS Secrets Manager Datenbankschlüssel ... 36
Schritt 5: Fügen Sie dem Secret den SSLMode hinzu ... 37
Schritt 6: Erstellen Sie eine Laufzeitumgebung ... 38
Schritt 7: Erstellen Sie eine Anwendung ... 43
Schritt 8: Stellen Sie eine Anwendung bereit ... 49
Schritt 9: Datensätze importieren ... 51
Schritt 10: Starten Sie eine Anwendung ... 57
Schritt 11: Connect zur CardDemo CICS-Anwendung her ... 58
Bereinigen von -Ressourcen ... 66
Nächste Schritte ... 67
Lebenszyklus der Komponenten ... 68
Überblick über den Lebenszyklus der Komponenten ... 68
Versions-Upgrade ... 70
AWS Überblick über die Veröffentlichung von Mainframe Modernization Refactor with Blu Age AWS ... 70
Verwaltete Anwendungen ... 72
Erstellen Sie AWS Ressourcen für eine migrierte Anwendung ... 73
Erforderliche Berechtigungen ... 73
Amazon-S3-Bucket ... 74
Datenbank ... 74
AWS Key Management Service Schlüssel ... 75
AWS Secrets Manager geheim ... 75

Erstellen einer Anwendung ..... 76
Erstellen einer Anwendung ..... 76
Stellen Sie eine Anwendung bereit ..... 78
Stellen Sie eine Anwendung bereit ..... 78
Aktualisieren einer Anwendung ..... 79
Aktualisieren einer Anwendung ..... 79
Löschen einer Anwendung ..... 80
Löschen einer Anwendung ..... 80
Reichen Sie Batch-Jobs für Bewerbungen ein ..... 81
Reichen Sie einen Batch-Job ein ..... 82
Starten Sie einen Batch-Job neu ..... 82
Stornieren Sie Batch-Jobs für Bewerbungen ..... 84
Stornieren Sie einen Batch-Job ..... 84
Datensätze für Anwendungen importieren ..... 85
Importieren Sie einen Datensatz ..... 85
Exportieren Sie Datensätze für Anwendungen ..... 86
Exportieren Sie einen Datensatz ..... 86
Transaktionen für Anwendungen verwalten ..... 87
Transaktionen für Anwendungen verwalten ..... 87
Konfigurieren Sie die von Rocket Software verwaltete Anwendung ..... 89
Unterstützte Integrationen von Drittanbietern für Rocket Software ..... 89
Konfigurieren Sie die verwaltete AWS Blu Age-Anwendung ..... 91
Struktur der von AWS Blu Age verwalteten Anwendungen ..... 92
Zugriff auf Dienstprogramme für verwaltete Anwendungen konfigurieren ..... 94
Konfigurieren Sie zusätzliche Eigenschaften für die verwaltete Anwendung ..... 105
Referenz zur Anwendungsdefinition ..... 127
Allgemeiner Kopfbereich ..... 128
Überblick über den Definitionsbereich ..... 130
AWS Beispiel für eine Blu Age-Anwendungsdefinition ..... 130
AWS Einzelheiten zur Blu Age-Definition ..... 131
Anwendungsdefinition von Rocket Software ..... 137
Einzelheiten zur Definition von Rocket Software ..... 139
Referenz zur Definition von Datensätzen ..... 150
Gemeinsame Eigenschaften ..... 151
Beispiel für ein Datensatz-Anforderungsformat für VSAM ..... 153
Beispiel für ein Anforderungsformat für einen Datensatz für die GDG-Basis ..... 155

Beispiel für ein Anforderungsformat für Datensätze für PS- oder GDG-Generierungen ..... 156
Beispiel für ein Anforderungsformat für einen Datensatz für PO ..... 157
Verwaltete Laufzeitumgebungen ..... 160
Erstellen Sie eine Laufzeitumgebung ..... 161
Erstellen Sie eine Laufzeitumgebung ..... 161
Aktualisieren Sie eine Laufzeitumgebung ..... 165
Aktualisieren Sie eine Laufzeitumgebung ..... 165
Wartungsfenster ..... 166
Stoppen Sie eine Laufzeitumgebung ..... 166
Beenden Sie eine Laufzeitumgebung ..... 167
Starten Sie eine Laufzeitumgebung neu ..... 168
Starten Sie eine Laufzeitumgebung neu ..... 168
Löschen Sie eine Laufzeitumgebung ..... 169
Löschen Sie eine Laufzeitumgebung ..... 169
Testen von Anwendungen ..... 170
Was sind Anwendungstests ..... 170
Verwenden Sie Application Testing zum ersten Mal? ..... 171
Vorteile von Application Testing ..... 171
Integration mit AWS CloudFormation ..... 172
Wie funktioniert das Testen von Anwendungen ..... 173
Zugehörige Services ..... 4
Zugriff auf Anwendungstests ..... 174
Preise für Anwendungstests ..... 175
Konzepte für das Testen von Anwendungen ..... 175
Testfall ..... 176
Testsuite ..... 176
Konfiguration der Testumgebung ..... 177
Hochladen ..... 177
Erneut abspielen ..... 177
Compare ..... 177
Datenbankvergleiche ..... 177
Vergleiche von Datensätzen ..... 178
Status des Vergleichs ..... 179
Regeln für die Gleichwertigkeit ..... 179
Vergleich von Datensätzen im Endzustand ..... 179
Vergleiche zwischen Datenbanken zum Stand und zum Fortschritt ..... 180

Funktionale Äquivalenz (FE) ..... 180
Online-Bildschirmvergleiche von 3270 ..... 180
Daten erneut abspielen ..... 180
Referenzdaten ..... 181
Hochladen, Wiedergeben und Vergleichen ..... 181
Unterschiede ..... 182
Äquivalenzen ..... 182
Quellanwendung ..... 182
Zielanwendung ..... 182
Voraussetzungen für Anwendungstests ..... 183
Konsolen-Workflows beim Testen von Anwendungen ..... 183
Erstellen Sie Testfälle beim Testen von Anwendungen ..... 184
Erstellen Sie Testsuiten in Application Testing ..... 186
Erstellen Sie Testumgebungskonfigurationen in Application Testing ..... 189
Tutorial: Richten Sie die CardDemo Anwendung in Application Testing ein ..... 191
Voraussetzungen ..... 191
Schritt 1: Bereiten Sie die Einrichtung vor CardDemo ..... 191
Schritt 2: Erstellen Sie alle erforderlichen Ressourcen ..... 192
Schritt 3: Stellen Sie die Anwendung bereit und starten Sie sie ..... 193
Schritt 4: Erste Daten importieren ..... 193
Schritt 5: Connect zur CardDemo Anwendung her ..... 194
Tutorial: Auf Blu Age erneut abspielen und vergleichen mit AWS CardDemo ..... 195
Schritt 1: Besorgen Sie sich AWS Blu Age EC2 Amazon Amazon Machine Image (AMI) ..... 196
Schritt 2: Starten Sie eine EC2 Amazon-Instance mit dem AWS Blu Age AMI ..... 196
Schritt 3: Laden Sie CardDemo abhängige Dateien auf S3 hoch ..... 198
Schritt 4: Datenbanken laden und die Anwendung initialisieren CardDemo ..... 198
Schritt 5: Starten Sie AWS Blu Age Runtime CloudFormation ..... 200
Schritt 6: Testen der AWS Blu Age EC2 Amazon-Instanz ..... 203
Schritt 7: Überprüfen Sie, ob die vorherigen Schritte korrekt abgeschlossen wurden ..... 204
Schritt 8: Erstellen Sie den Testfall ..... 205
Schritt 9: Erstellen Sie eine Testsuite ..... 205
Schritt 10: Erstellen Sie eine Testumgebungskonfiguration ..... 206
Schritt 11: Laden Sie Ihre Eingabedaten in die Testsuite hoch ..... 206
Schritt 12: Wiedergeben und vergleichen ..... 207
Unterstützte Codepages für Datensätze beim Testen von Anwendungen ..... 207
Datenschutz beim Testen von Anwendungen ..... 218

Daten, die im Rahmen von AWS Mainframe Modernization Application Testing gesammelt wurden ... 219
Datenverschlüsselung im Ruhezustand für den AWS Mainframe Modernization Application Testing ... 220
Einen kundenverwalteten Schlüssel erstellen ... 221
Angabe eines vom Kunden verwalteten Schlüssels für das Testen von AWS Mainframe-Modernisierungsanwendungen ... 223
AWS Verschlüsselungskontext für Mainframe-Modernisierung, Anwendungstests ... 223
Überwachen Ihrer Verschlüsselungsschlüssel ... 224
Verschlüsselung während der Übertragung ... 224
So funktionieren Anwendungstests mit IAM ... 225
Identitätsbasierte Richtlinien ... 226
Ressourcenbasierte Richtlinien ... 226
Richtlinienaktionen ... 227
Richtlinienressourcen ... 228
Bedingungsschlüssel für die Richtlinie ... 230
ACLs ... 231
ABAC ... 231
Temporäre Anmeldeinformationen ... 232
Forward Access Sessions (FAS) ... 232
Servicerollen ... 233
Service-verknüpfte Rollen ... 233
AWS Blu Age Refactoring ... 234
AWS Veröffentlichungen von Blu Age ... 235
AWS Blu-Age-Versionierung ... 236
AWS Optionen für Blu Age Runtime ... 237
AWS Versionshinweise zu Blu Age ... 240
AWS Sicherheitslücken in Blu Age ... 353
AWS Blu Age aktualisieren ... 354
AWS Lebenszyklus von Blu Age ... 357
AWS Konzepte von Blu Age Runtime ... 358
Architektur auf hohem Niveau ... 358
Modernisierte Anwendungsstruktur ... 363
Verstehen Sie Datenvereinfacher ... 400
AWS Blu Age Blusam ... 408
Verfügbare Programme in der Utility-Webanwendung ... 432

Blusam Verwaltungskonsole ... 434
AWS Blu Age Runtime-Konfiguration ... 475
Grundlagen der Anwendungskonfiguration ... 476
Vorrang der Anwendung ... 478
JNDI für Datenbanken ... 478
AWS Die Geheimnisse von Blu Age Runtime ... 479
Andere Dateien (Groovy, SQL usw.) ... 491
Zusätzliche Webanwendung ... 492
Eigenschaften aktivieren ... 492
Verfügbare Redis-Cache-Eigenschaften ... 585
Konfigurieren Sie die Sicherheit für Gapwalk-Anwendungen ... 602
AWS Blu Age Runtime APIs ... 618
Endpunkte zum Bauen URLs ... 618
Endpunkte für die Gapwalk-Anwendung ... 619
REST-Endpunkte der Blusam-Anwendungskonsole ... 640
JICS-Anwendungskonsole verwalten ... 663
Datenstrukturen ... 684
AWS Blu Age Runtime einrichten (nicht verwaltet) ... 693
AWS Voraussetzungen für Blu Age Runtime ... 694
Einführung in AWS Blu Age Runtime ... 695
Anforderungen an die Einrichtung der Infrastruktur ... 700
AWS Artefakte von Blu Age Runtime ... 707
Stellen Sie AWS Blu Age Runtime auf Amazon bereit EC2 ... 710
Stellen Sie AWS Blu Age Runtime auf Amazon ECS und Amazon EKS bereit ... 721
Testen Sie die PlanetsDemo Anwendung ... 729
Ändern Sie den Quellcode mit Blu Age Developer IDE ... 733
Tutorial: AppStream 2.0 für AWS Blu Age Developer IDE einrichten ... 733
Tutorial: Verwenden Sie AWS Blu Age Developer auf AppStream 2.0 ... 739
AWS Blu Age Häufig gestellte Fragen ... 756
Allgemeines ... 756
AWS Blu Age Runtime ... 759
Daten ... 768
Transformation ... 770
Bereitstellung ... 770
Replatforming durch Rocket Software ... 774
Rocket Software einrichten (bei Amazon EC2) ... 774

Voraussetzungen für Rocket Software (bei Amazon EC2) ... 775
Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3 ... 775
Fordern Sie die Aktualisierung der Zulassungsliste für das Konto an ... 778
Erstellen Sie die AWS Identity and Access Management Rolle ... 779
Gewähren Sie License Manager die erforderlichen Berechtigungen ... 787
Abonnieren Sie die Amazon Machine Images ... 788
Starten Sie eine Rocket Software-Instanz ... 791
Subnetz oder VPC ohne Internetzugang ... 798
Richten Sie die AppStream 2.0-Automatisierung ein ... 804
Richten Sie die Automatisierung beim Sitzungsstart ein ... 805
Richten Sie die Automatisierung am Ende der Sitzung ein ... 805
Datensätze in Enterprise Developer als Tabellen anzeigen ... 805
Voraussetzungen ... 806
Schritt 1: ODBC-Verbindung zum Rocket Software-Datenspeicher (Amazon RDS-Datenbank) einrichten ... 806
Schritt 2: Erstellen Sie die Datei MFDBFH.cfg ... 809
Schritt 3: Erstellen Sie eine Strukturdatei (STR) für Ihr Copybook-Layout ... 810
Schritt 4: Erstellen Sie eine Datenbankansicht mithilfe der Strukturdatei (STR) ... 812
Schritt 5: Datensätze von Rocket Software (ehemals Micro Focus) als Tabellen und Spalten anzeigen ... 813
Bearbeiten Sie Datensätze mit den Datendatei-Tools in Enterprise Developer ... 814
Voraussetzungen ... 815
Starten Sie die Data File Tools von Rocket Software (ehemals Micro Focus) ... 815
Bearbeiten Sie die in der MFDBFH-Datenbank gespeicherten VSAM-Datensätze ... 816
Bearbeiten Sie Nicht-VSAM-Datensätze, die in der MFDBFH-Datenbank gespeichert sind ... 819
Bearbeiten Sie im Dateisystem gespeicherte VSAM- und Nicht-VSAM-Datensätze (EFS/FSx) ... 821
Tutorials für Rocket Software ... 822
Tutorial: Richten Sie den Build für die BankDemo Beispielanwendung ein ... 822
Tutorial: Einrichten CI/CD Pipeline mit Rocket Enterprise Developer ... 833
Tutorial: AppStream 2.0 für Enterprise Analyzer und Enterprise Developer einrichten ... 859
Tutorial: Verwenden Sie Vorlagen mit Rocket Enterprise Developer ... 869
Tutorial: Enterprise Analyzer einrichten ... 880
Tutorial: Enterprise Developer einrichten ... 891
Batch-Dienstprogramme ... 897
Binärer Standort ... 898

M2SFTP-Batch-Hilfsprogramm ... 898
M2WAIT-Batch-Hilfsprogramm ... 905
TXT2PDF-Batch-Hilfsprogramm ... 907
M2DFUTIL-Batch-Hilfsprogramm ... 913
M2RUNCMD Batch-Hilfsprogramm ... 921
Übertragung von Dateien ... 925
Was ist Dateiübertragung ... 925
Vorteile von AWS Mainframe Modernization File Transfer ... 926
So funktioniert AWS Mainframe Modernization File Transfer ... 926
Installieren Sie einen File Transfer Agent ... 927
Schritt 1: Erstellen Sie einen ZfS-Datensatz für den M2-Agenten ... 928
Schritt 2: Formatieren Sie den Datensatz als ZFS ... 928
Schritt 3: Mounten Sie das Dateisystem ... 928
Schritt 4: Überprüfen Sie den Mount ... 929
Schritt 5: Geben Sie OMVS ein ... 929
Schritt 6: Legen Sie die Umgebungsvariable für das Installationsverzeichnis des Agenten fest ... 929
Schritt 7: Legen Sie die Umgebungsvariable für das Arbeitsverzeichnis fest ... 929
Schritt 8: Erstellen Sie das Arbeitsverzeichnis ... 930
Schritt 9: Kopieren Sie die Tar-Datei des Agenten und das Arbeitsverzeichnis ... 930
Schritt 10: Beenden Sie die Agenteninstallation ... 930
Konfigurieren Sie einen Dateiübertragungsagenten ... 931
Schritt 1: Berechtigungen konfigurieren und Task Control (STC) starten ... 932
Schritt 2: Amazon S3 S3-Buckets erstellen ... 933
Schritt 3: Erstellen Sie einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung ... 933
Schritt 4: Erstellen Sie ein AWS Secrets Manager Geheimnis für die Mainframe-Anmeldeinformationen ... 934
Schritt 5: Erstellen Sie eine IAM-Richtlinie ... 936
Schritt 6: Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten ... 937
Schritt 7: Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll ... 939
Schritt 8: Agentenkonfiguration ... 940
Datenübertragungsendpunkte erstellen ... 942
Datenübertragungsendpunkte erstellen ... 943
Erstellen Sie Übertragungsaufgaben ... 945
Übertragungsaufgaben erstellen ... 945

Übertragungsaufgaben anzeigen ... 949
Tutorial: Erste Schritte mit File Transfer ... 949
Übersicht ... 950
Schritt 1: Übertragen Sie das Tar-Paket der Agent-Binärdateien von AWS auf die logische Mainframe-Partition ... 950
Schritt 2: Konfigurieren Sie den File Transfer Agent auf dem Quell-Mainframe ... 950
Schritt 3: Erstellen Sie einen Datenübertragungsendpunkt ... 951
Schritt 4: Erstellen Sie eine Übertragungsaufgabe ... 951
Schritt 5: Status der Übertragungsaufgabe einsehen ... 951
Unterstützte Quell- und Zielcodeseiten ... 951
Mainframe-Datensatztypen ... 951
Unterstützte Codepages ... 951
AWS Transform für Mainframe ... 953
Wichtigste Vorteile ... 953
Komplettlösung zur Transformation der Mainframe-Anwendungskonsole ... 954
Datenreplikation mit Precisely ... 955
Voraussetzungen ... 955
Abonnieren Sie das Amazon Machine Image ... 955
Starten Sie die AWS Mainframe Modernization Datenreplikation mit Precisely ... 956
Eine IAM-Richtlinie erstellen ... 957
Erstellen einer IAM-Rolle ... 958
Hängen Sie die IAM-Rolle an die EC2 Amazon-Instance an ... 958
Assembler-Konvertierung mit mLogica ... 960
Was ist Assembler-Konvertierung mit mLogica ... 960
Compiler für die Codekonvertierung ... 961
Architektur für die Codekonvertierung ... 961
Automatisierungsansatz ... 962
Sicherheit ... 962
Weitere Ressourcen ... 963
Verstehen Sie die Abrechnung der Codekonvertierung ... 963
Abrechnung und Umfang der Codekonvertierung ... 963
Konzepte zur Codekonvertierung ... 966
Umgang mit Makros ... 966
Codepages (EBCDIC oder ASCII) ... 966
CodeBuild ... 966
Komponenten und Prozesse verstehen ... 967

AWS Mainframe Modernization Container ... 967
S3-Projekt-Bucket ... 968
Speicherorte der Protokolldateien ... 968
Prozessübersicht ... 969
Tutorial: Code von Assembler nach COBOL konvertieren ... 969
Voraussetzungen ... 970
Schritt 1: Teilen Sie die Build-Assets mit AWS-Konto ... 970
Schritt 2: Amazon S3 S3-Buckets erstellen ... 971
Schritt 3: IAM-Richtlinie erstellen ... 971
Schritt 4: Erstellen Sie eine IAM-Rolle ... 973
Schritt 5: Hängen Sie die IAM-Richtlinie an die IAM-Rolle an ... 974
Schritt 6: Erstellen Sie das CodeBuild Projekt ... 974
Schritt 7: Definieren Sie das Projekt und laden Sie den Quellcode hoch ... 981
Schritt 8: Führen Sie die Analyse durch und verstehen Sie die Berichte ... 982
Schritt 9: Führen Sie die Codekonvertierung aus ... 984
Schritt 10: Überprüfen Sie die Codekonvertierung ... 988
Schritt 11: Laden Sie den konvertierten Code herunter ... 989
Bereinigen von -Ressourcen ... 990
Integration von Charon ... 991
Einführung in Charon-SSP ... 991
Unterstützte Gastbetriebssysteme ... 993
Voraussetzungen für die Charon-SSP Cloud-Instanz ... 994
Voraussetzungen für Instance ... 995
Erstellen und Konfigurieren einer AWS Cloud-Instanz für Charon (neue GUI) ... 997
Allgemeine Voraussetzungen ... 997
Verwenden von AWS Management Console , um eine neue Instanz zu starten ... 998
Umgestaltung der Plattform mit NTT DATA ... 1004
Voraussetzungen ... 1004
Abonnieren Sie das Amazon Machine Image ... 1004
Starten Sie die AWS Mainframe Modernization Replatform mit der NTT DATA-Instanz ... 1005
Erste Schritte mit NTT Data ... 1006
Tutorial: Anwendung auf NTT DATA bereitstellen CardDemo ... 1007
Flussdiagramm für die Bereitstellung ... 1008
Voraussetzungen ... 1008
Schritt 1: Bereiten Sie die Umgebung vor ... 1009
Schritt 2: Erstellen Sie eine TPE-Region ... 1010

Schritt 3: Erstellen Sie den BPE-Knoten und das Subsystem ... 1011
Schritt 4: Anwendung kompilieren und bereitstellen CardDemo ... 1020
Schritt 5: BPE- und TPE-Katalog importieren ... 1022
Schritt 6: Starten und verbinden Sie TPE mit BPE ... 1022
Schritt 7: Führen Sie die Anwendung aus CardDemo ... 1023
Fehlerbehebung ... 1029
Sicherheit ... 1031
Datenschutz ... 1032
Daten, die AWS Mainframe Modernization sammelt ... 1033
Datenverschlüsselung im Ruhezustand für den AWS Mainframe-Modernisierungsservice . 1034
So verwendet AWS Mainframe Modernization Zuschüsse in AWS KMS ... 1037
Einen kundenverwalteten Schlüssel erstellen ... 1039
Angabe eines vom Kunden verwalteten Schlüssels für die AWS Mainframe-Modernisierung ... 1040
AWS Verschlüsselungskontext für die Mainframe-Modernisierung ... 1041
Überwachen Ihrer Verschlüsselungsschlüssel ... 1043
Weitere Informationen ... 1059
Verschlüsselung während der Übertragung ... 1059
Identitäts- und Zugriffsverwaltung ... 1059
Zielgruppe ... 1060
Authentifizierung mit Identitäten ... 1061
Verwalten des Zugriffs mit Richtlinien ... 1065
So funktioniert die AWS Mainframe-Modernisierung mit IAM ... 1068
Beispiele für identitätsbasierte Richtlinien ... 1084
Fehlerbehebung ... 1087
Verwenden von serviceverknüpften Rollen ... 1089
Compliance-Validierung ... 1093
Ausfallsicherheit ... 1094
Sicherheit der Infrastruktur ... 1094
AWS PrivateLink ... 1095
Überlegungen ... 1095
Erstellen eines Schnittstellenendpunkts ... 1095
Erstellen einer Endpunktrichtlinie ... 1096
Überwachen ... 1098
Überwachung mit CloudWatch ... 1098
Metriken zur Laufzeitumgebung ... 1099

Anwendungsmetriken ..... 1100
Dimensionen ..... 1106
Protokollierung von CloudTrail -API-Aufrufen mit ..... 1106
AWS Informationen zur Mainframe-Modernisierung finden Sie unter CloudTrail ..... 1107
Grundlegendes zu den Einträgen in der AWS Mainframe-Modernisierung in der ..... 1108
Fehlerbehebung in M2 ..... 1110
Fehler zur Fehlerbehebung: Beim Warten auf die Entsperrung des Datensatznamens wurde eine Zeitüberschreitung überschritten ..... 1110
Häufige Ursache ..... 1111
Auflösung ..... 1111
Erzwingen Sie das Lösen des Schlosses ..... 1111
Konfigurieren Sie den auto Reparaturmechanismus von Blusam ..... 1112
Blusam Locks Manager ..... 1113
Fehler zur Fehlerbehebung: Auf eine Anwendungs-URL kann nicht zugegriffen werden ..... 1114
Häufige Ursache ..... 1114
Auflösung ..... 1114
Fehlerbehebung: AWS Blu Insights lässt sich nicht von der Konsole aus öffnen ..... 1115
Häufige Ursache ..... 1116
Auflösung ..... 1116
Fehler zur Fehlerbehebung: Die Umgebung ist fehlerhaft ..... 1116
Häufige Ursache ..... 1117
Auflösung ..... 1117
Behebung von Lizenzproblemen für Rocket Software ..... 1118
Stellen Sie sicher, dass die EC2 Amazon-Instance die IAM-Lizenzierungsrolle hat ..... 1118
Verwenden Sie den Erreichbarkeitsanalysator ..... 1119
Führen Sie den License-Daemon aus ..... 1119
Lizenzprobleme mit Enterprise Server oder Enterprise Build Tools unter Linux nach dem Patchen des Betriebssystems ..... 1120
Dokumentverlauf ..... 1122
..... mcxxx

# Was ist AWS Mainframe-Modernisierung?

AWS Mainframe-Modernisierung hilft Ihnen dabei, Ihre Mainframe-Anwendungen zu AWS verwalteten Laufzeitumgebungen zu modernisieren. 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 entwickeln oder PL/I aktualisieren und eine automatisierte Pipeline für die kontinuierliche Integration und Bereitstellung der Anwendungen implementieren. (CI/CD) 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 die AWS Mainframe-Modernisierung
- Verwenden Sie Mainframe Modernization zum ersten Mal? AWS
- Preise für AWS die Mainframe-Modernisierung


> **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 () und Anwendungssoftware-Stacks zu modernisieren. UIs

  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 Unternehmens-Workloads mit selbstreparierender Rechenleistung und automatisierter Skalierung am Laufen zu halten.
- Kontinuierliche Integration und Bereitstellung(CI/CD): Die CI/CD Funktion der AWS Mainframe-Modernisierung unterstützt Anwendungsentwicklungsteams dabei, Codeänderungen häufiger und zuverlässiger durchzuführen. Dadurch wird die Migrationsgeschwindigkeit beschleunigt, die Qualität erhöht und der Aufwand time-to-market für die Veröffentlichung neuer Geschäftsfunktionen reduziert.
- 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: 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 finden Sie unter Funktionen der AWS Mainframe-Modernisierung.

# Muster

Das automatisierte Refactoring-Pattern, 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 entsteht eine mehrstufige Anwendung mit einem Angular-basierten Frontend, einem API-fähigen Java-Backend und einer Datenschicht, 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 Enterprise Suite von Rocket Software (ehemals Micro Focus) 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.

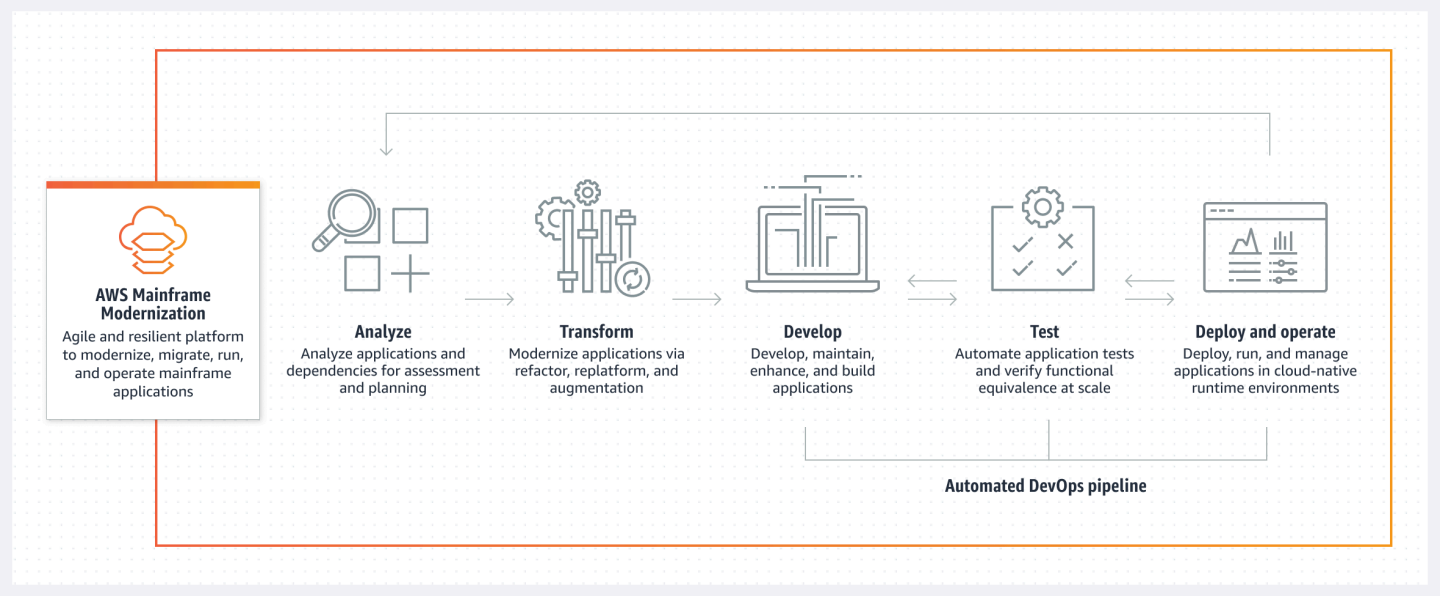
# Wie fange ich mit AWS der Mainframe-Modernisierung an

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 einrichten oder das, Tutorial: Verwaltete Laufzeit für Rocket Software (ehemals Micro Focus) einrichten 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 einrichten, um auf die AWS Blu Age Developer IDE oder die Tools Rocket Enterprise Analyzer (ehemals Micro Focus Enterprise Analyzer) und Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer) zuzugreifen.

Die Tutorials und Beispielanwendungen geben Ihnen nur einen Eindruck davon, was 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.

Das folgende Diagramm zeigt den Arbeitsablauf des AWS Mainframe Modernization Service zur Analyse, Transformation, Entwicklung, Erprobung, Bereitstellung und Betrieb von Mainframe-Anwendungen.




# 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 Rocket Enterprise Analyzer und Rocket Enterprise Developer

- AWS CloudFormation für die automatisierte DevOps Pipeline, mit der Sie Ihre migrierten Anwendungen einrichten CI/CD können
- AWS Migration Hub
- AWS DMS für die Migration Ihrer Datenbanken

## Zugriff auf die AWS Mainframe-Modernisierung

Derzeit können Sie über die Konsole AWS unter 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 im. 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 mit der AWS Mainframe-Modernisierung
- Für die AWS Mainframe-Modernisierung gerüstet

## Preise für AWS die Mainframe-Modernisierung

AWS Bei der Mainframe-Modernisierung fallen Gebühren für die Nutzung von Instanzen an, 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 Sie werden 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.

# Für die AWS Mainframe-Modernisierung gerüstet

Bevor Sie AWS Mainframe Modernization nutzen können, müssen Sie oder Ihr Administrator sich für einen registrieren AWS-Konto, einen Benutzer mit Administratoreinstellungen erstellen und Ihre IAM-Benutzer schützen.

Themen


- Melden Sie sich für ein AWS-Konto
- Erstellen eines Benutzers mit Administratorzugriff


## Melden Sie sich 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/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Ein Teil des Anmeldevorgangs umfasst den Empfang eines Telefonanrufs oder einer Textnachricht und die Eingabe eines Bestätigungscodes auf der Telefontastatur.

   Wenn Sie sich für eine anmelden AWS-Konto, wird eine Root-Benutzer des AWS-Kontoserstellt. Der Root-Benutzer hat Zugriff auf alle AWS-Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

# Erstellen eines Benutzers 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 Consoleals 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 eines Benutzers 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 Administratorbenutzer im IAM Identity Center Benutzerzugriff.

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

Anmelden als Administratorbenutzer

- 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 Access-Portal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie im IAM-Identity-Center einen Berechtigungssatz, der den bewährten Vorgehensweisen für die Anwendung von geringsten Berechtigungen folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.
2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

# AWS Konzepte zur Mainframe-Modernisierung

AWS Mainframe-Modernisierung bietet Tools und Ressourcen, mit denen Sie Mainframe-Workloads migrieren, modernisieren und ausführen können. AWS Auf dieser Seite können Sie sich mit verschiedenen Konzepten der AWS Mainframe-Modernisierung vertraut machen, darunter Anwendungen, Modernisierung, Umgebungen, Replatforming, Refactoring und Runtime-Engines.

Themen


- Anwendung
- Definition der Anwendung
- Batch-Auftrag
- Konfiguration
- Dataset
- Umgebung
- Mainframe-Modernisierung
- Reise zur Migration
- Montagepunkt
- 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.

Wird versendet

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 älterer Anwendungen, der in der Regel aus den folgenden Phasen besteht: Bewertung, Mobilisierung, Migration und Modernisierung sowie Betrieb und Optimierung.

## Montagepunkt

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.

# 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 Rocket Software (ehemals 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 einen 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.

# Erste Schritte mit der AWS Mainframe-Modernisierung

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

Themen

- Tutorial: Verwaltete Laufzeit für AWS Blu Age einrichten
- Tutorial: Verwaltete Laufzeit für Rocket Software (ehemals Micro Focus) einrichten

Weitere Informationen finden Sie in den folgenden Tutorials.

- Tutorial: Den Build von Rocket Software (ehemals Micro Focus) für die BankDemo Beispielanwendung einrichten
- Tutorial: Einrichtung eines CI/CD Pipeline zur Verwendung mit Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer)

# Tutorial: Verwaltete Laufzeit für AWS Blu Age einrichten

Sie können eine modernisierte AWS Blu Age-Anwendung mit einer in diesem Tutorial angegebenen Demo-Anwendung in einer Laufzeitumgebung für die AWS Mainframe-Modernisierung 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

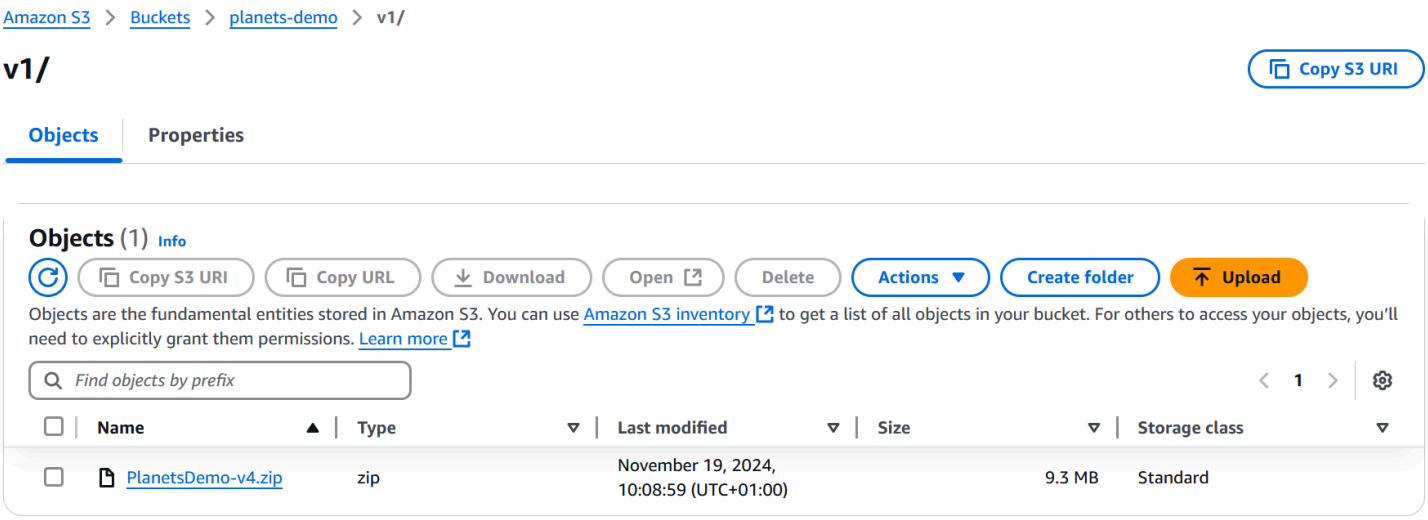
## Voraussetzungen

Um dieses Tutorial abzuschließen, laden Sie das Demo-Anwendungsarchiv PlanetsDemo -v4.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 planets-demo mit einem key prefix oder Ordner namens v1 und einem Archiv mit dem Namen. `planetsdemo-v4.zip`




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

Ändern Sie den `s3-bucket` Eintrag in den Namen der ZIP-Datei der Beispielanwendung (z. B.`planets-demo`) und den `app-location` Eintrag in den S3-Pfad, in dem Sie die Zip-Datei der Beispielanwendung gespeichert haben (z. B.`${s3-source}/PlanetsDemo-v4.zip`).

> **Note**
>
> Stellen Sie sicher, dass Sie die Anwendungsdefinitionsdatei lokal als Textdatei erstellen.

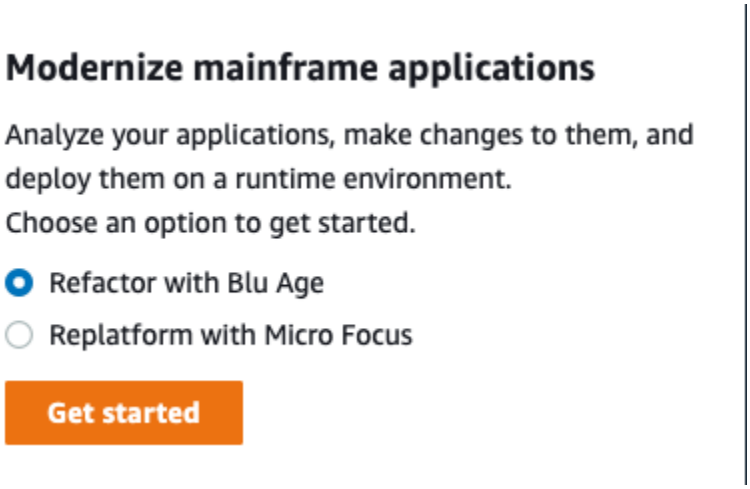
Weitere Informationen zur Anwendungsdefinition finden Sie unterAWS Beispiel für eine Blu-Age-Anwendungsdefinition.

## Schritt 3: Erstellen Sie eine Laufzeitumgebung

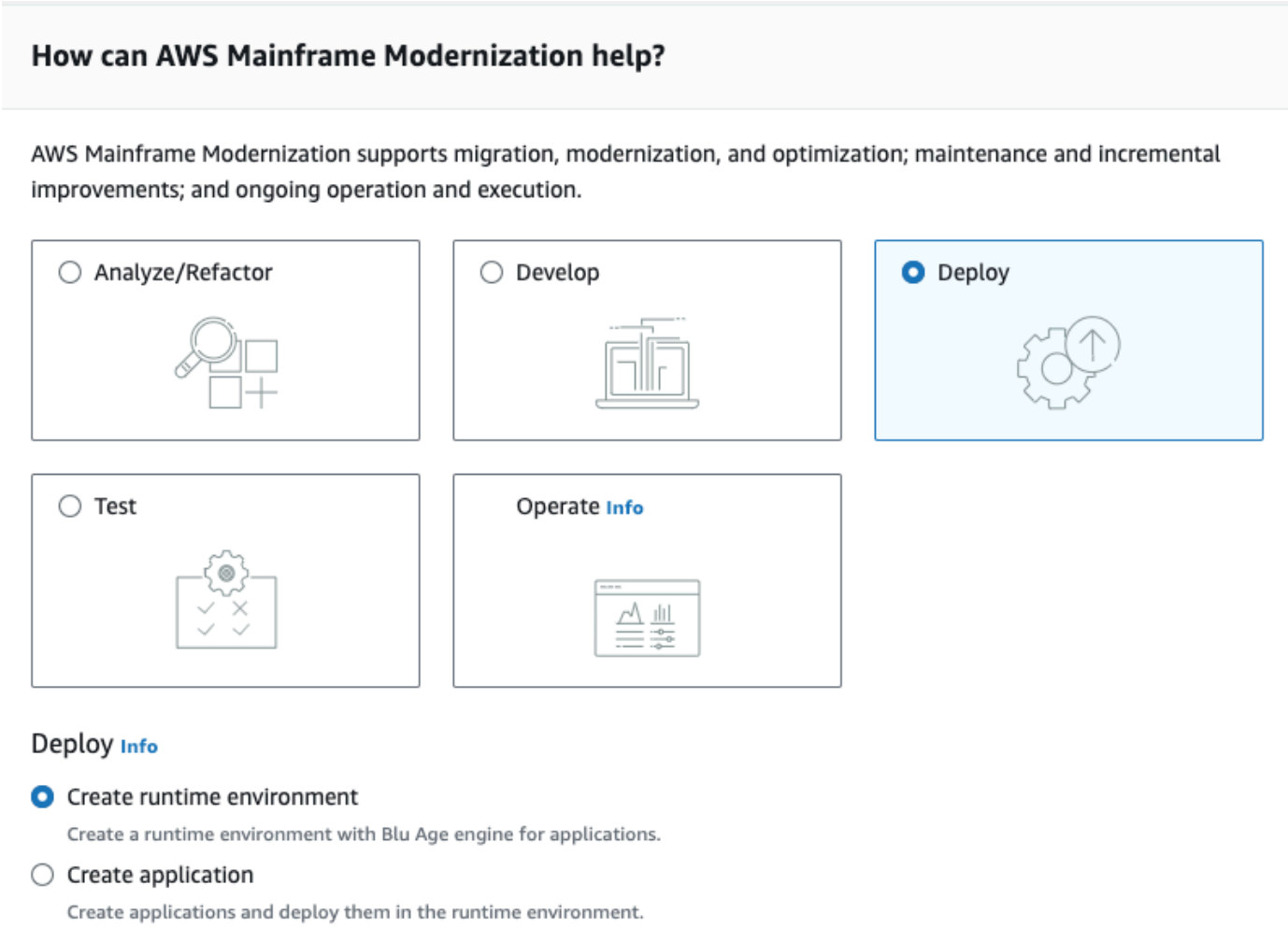
Gehen Sie wie folgt vor, um die Laufzeitumgebung für die AWS Mainframe-Modernisierung zu erstellen:

1. Öffnen Sie die AWS Mainframe Modernization Console.

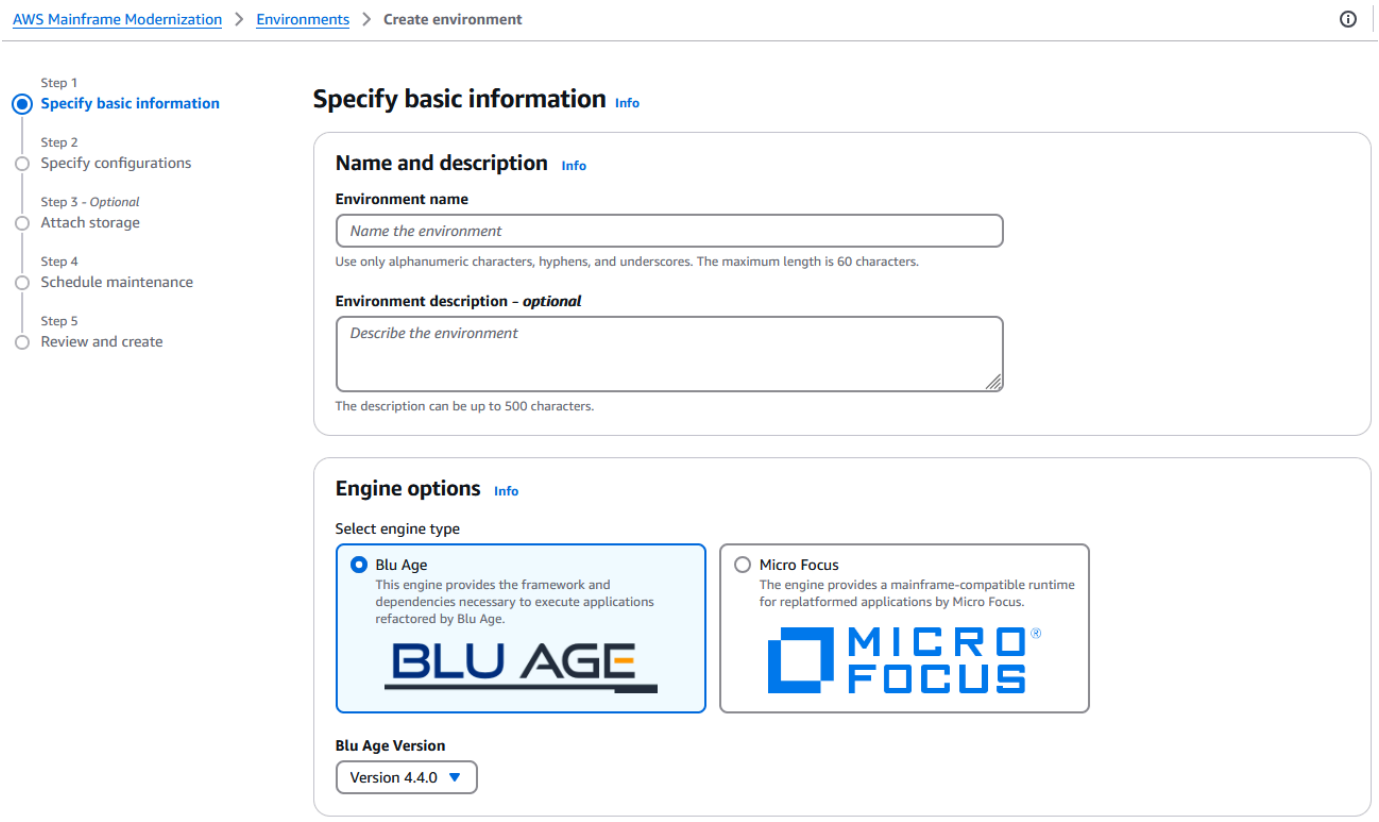
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Sie die Umgebung erstellen möchten. Dies AWS-Region muss mit der Region übereinstimmen, in der Sie den S3-Bucket erstellt haben. Schritt 1: Laden Sie die Demo-Anwendung hoch
3. Wählen Sie unter Mainframe-Anwendungen modernisieren die Option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



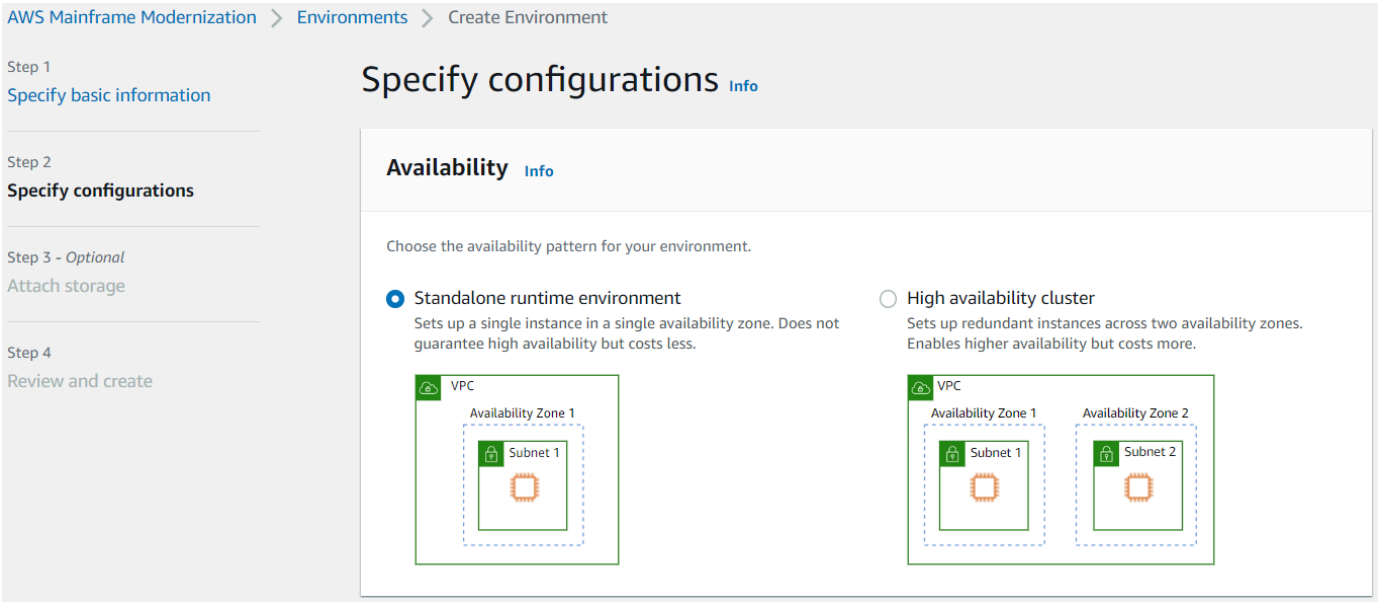

4. Wählen Sie unter Wie kann AWS Mainframe Modernization helfen? die Option Laufzeitumgebung bereitstellen und erstellen aus.




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 Age-Engine ausgewählt ist. Optional können Sie der erstellten Ressource Tags hinzufügen. Wählen Sie anschließend Weiter.

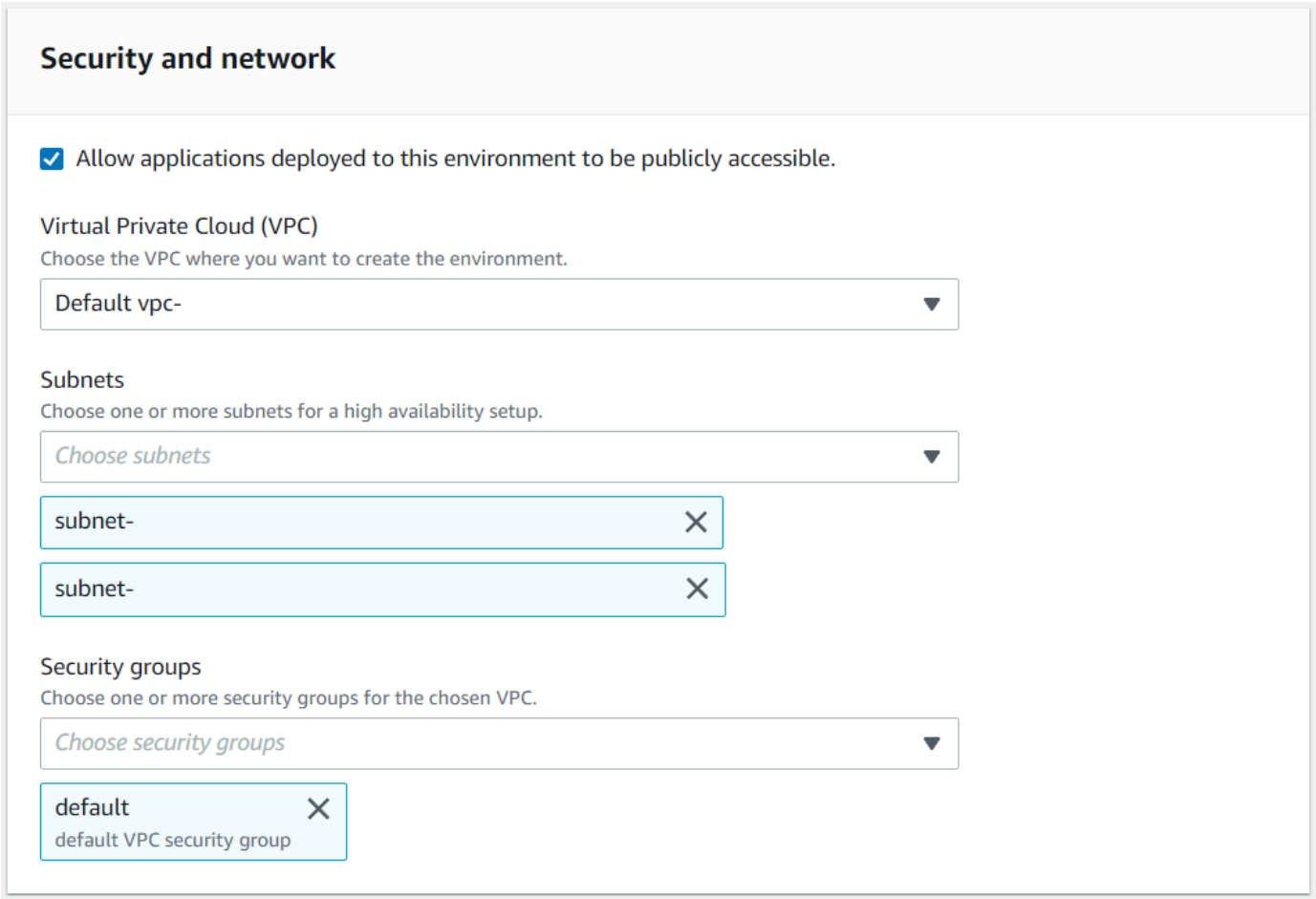



6. Wählen Sie auf der Seite „Konfigurationen angeben“ die Option Standalone-Laufzeitumgebung aus.




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

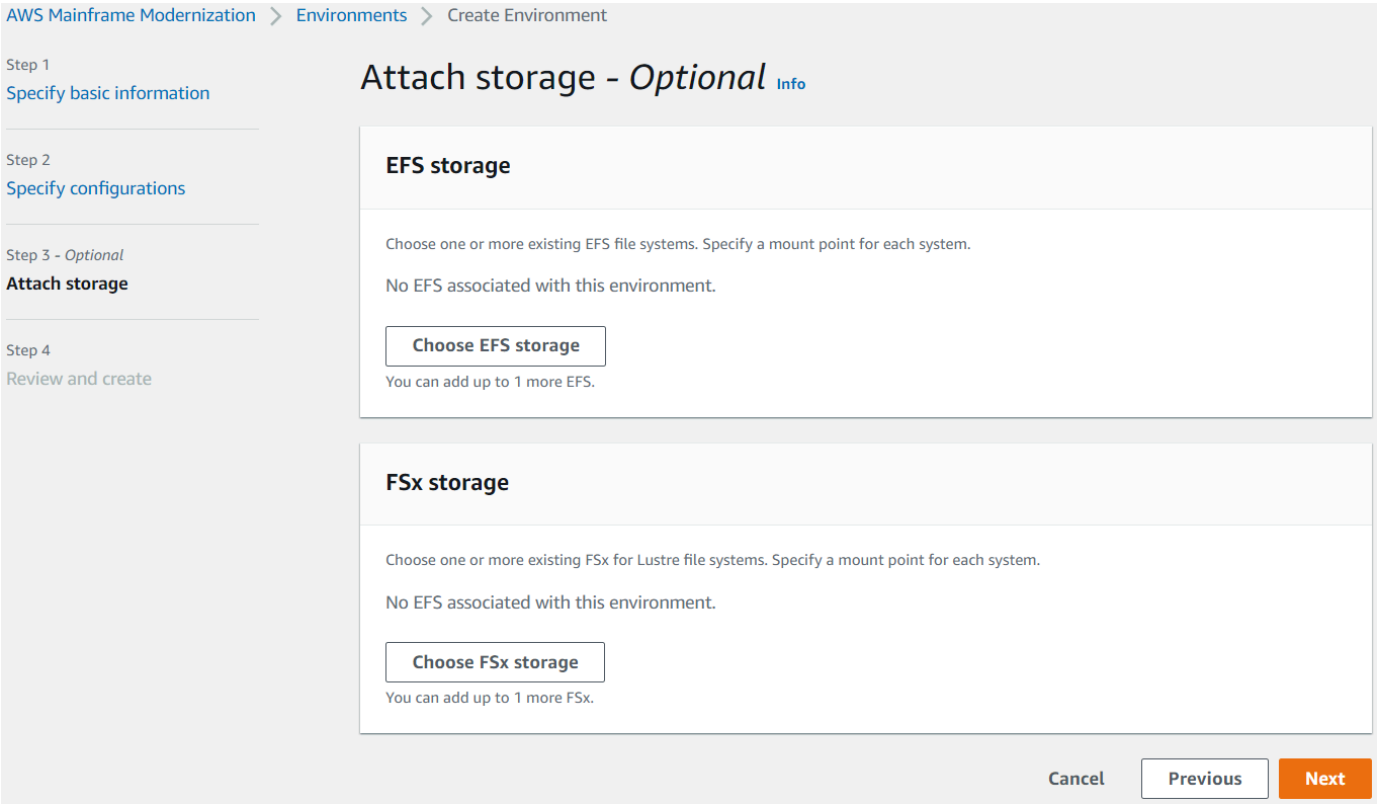
- Wählen Sie In dieser Umgebung bereitgestellte Anwendungen öffentlich zugänglich machen aus. Diese Option weist der Anwendung eine öffentliche IP-Adresse zu, sodass Sie von Ihrem Desktop aus darauf zugreifen können.
- Wählen Sie eine VPC aus. Sie können die Standardeinstellung verwenden.
- Wählen Sie zwei Subnetze aus. Stellen Sie sicher, dass die Subnetze die Zuweisung von öffentlichen IP-Adressen zulassen.
- Wählen Sie eine Sicherheitsgruppe aus. Sie können den Standard 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.




Wenn Sie von außerhalb der von Ihnen ausgewählten VPC auf die Anwendung zugreifen möchten, stellen Sie sicher, dass die Eingangsregeln für diese VPC ordnungsgemäß konfiguriert

sind. Weitere Informationen finden Sie unter Fehler zur Fehlerbehebung: Auf eine Anwendungs-URL kann nicht zugegriffen werden.

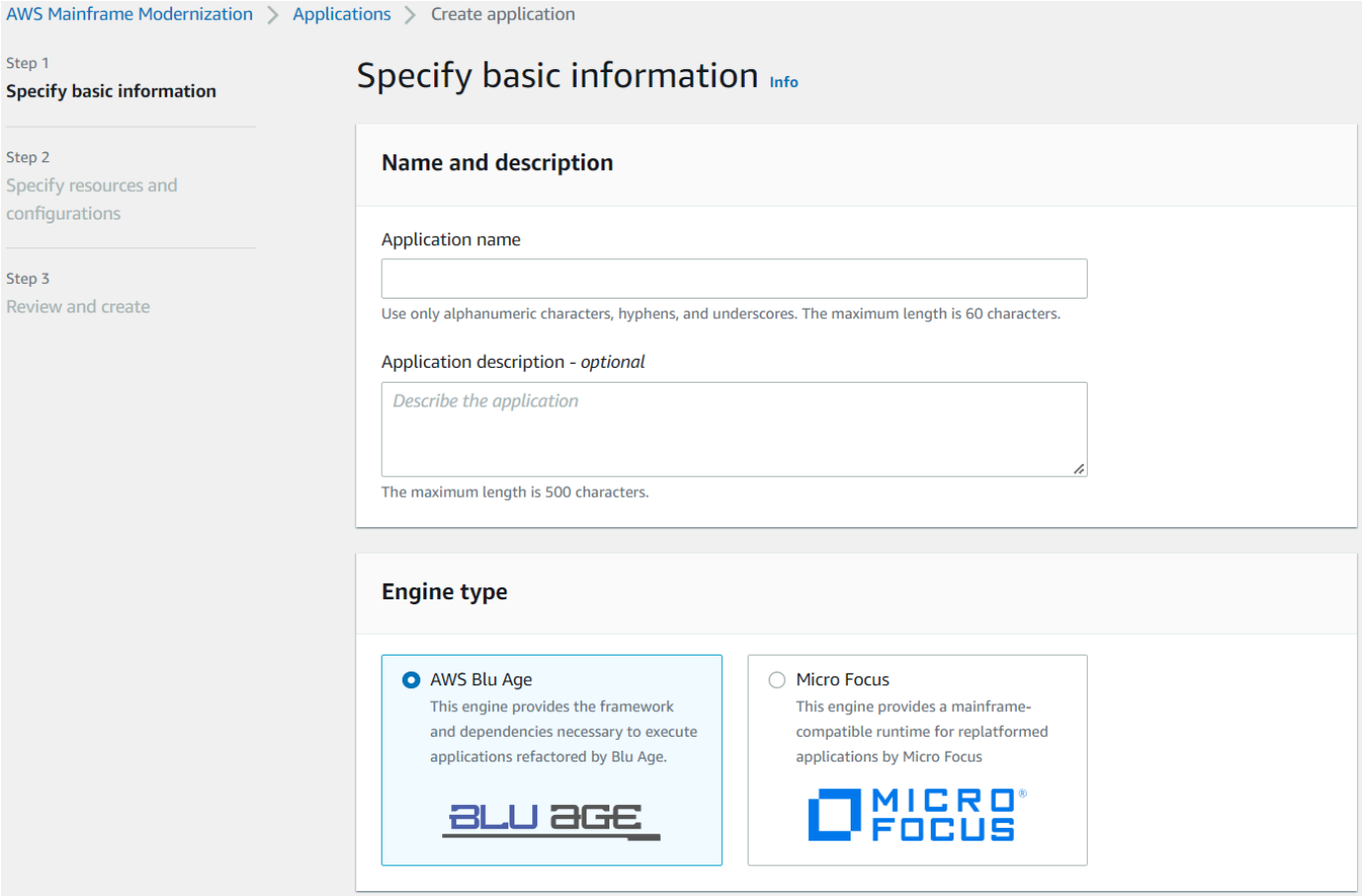
8. Wählen Sie Weiter aus.
9. Behalten Sie unter Speicher anhängen — Optional die Standardauswahl bei und wählen Sie Weiter.



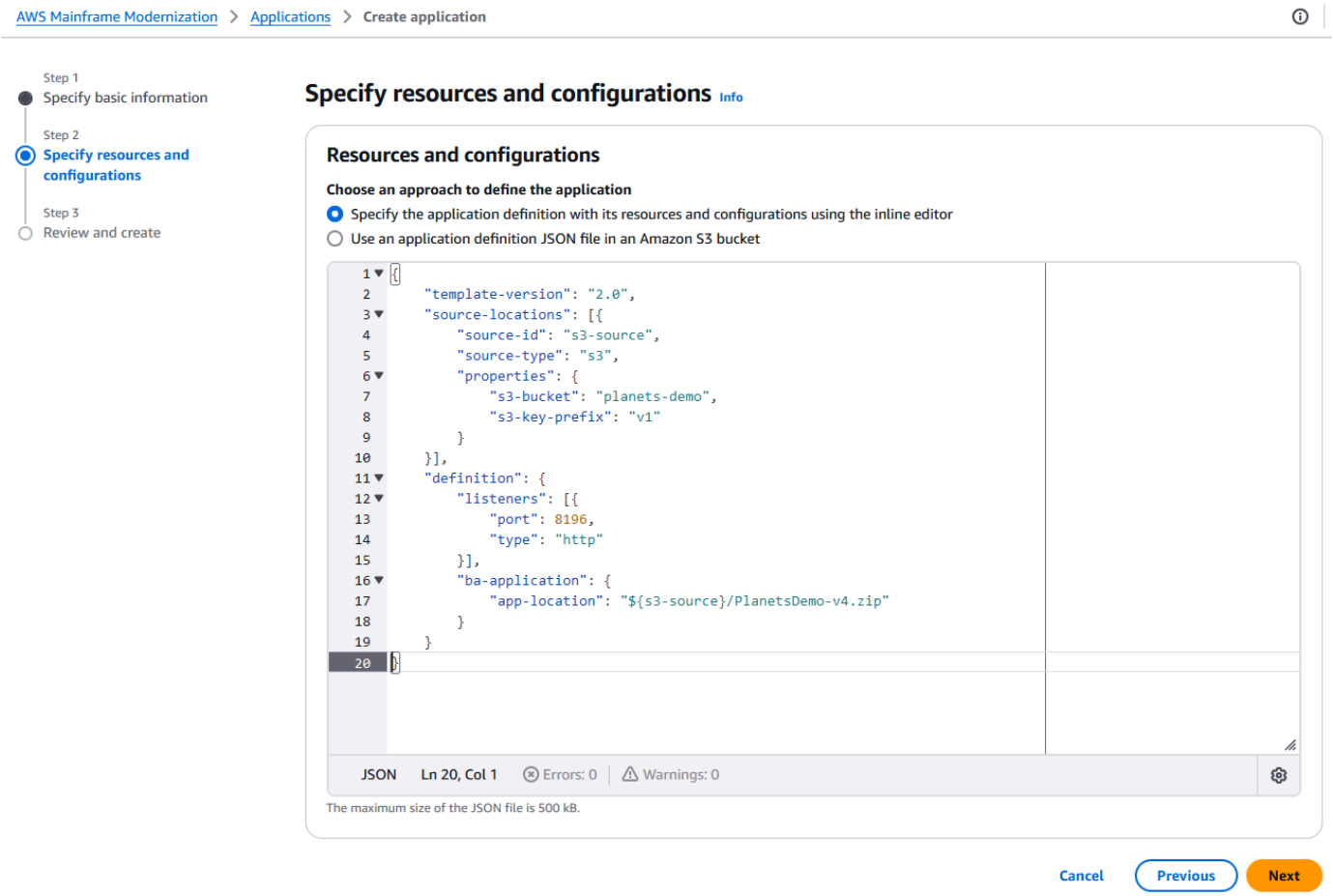

10. Wählen Sie unter Wartung planen die Option Keine Präferenz und dann Weiter aus.
11. Überprüfen Sie unter Überprüfen und erstellen die Informationen und wählen Sie dann Umgebung erstellen aus.

## 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 Age-Engine ausgewählt ist. Wählen Sie anschließend Weiter.




3. Kopieren Sie auf der Seite „Ressourcen und Konfigurationen angeben“ die aktualisierte JSON-Anwendungsdefinition, in der Sie sie erstellt haben, und fügen Sie sie einthe section called “Schritt 2: Erstellen Sie die Anwendungsdefinition”.




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

> **Note**
>
> Wenn Ihre Anwendungserstellung fehlschlägt, überprüfen Sie den eingegebenen S3-Pfad, da Groß- und Kleinschreibung beachtet wird.

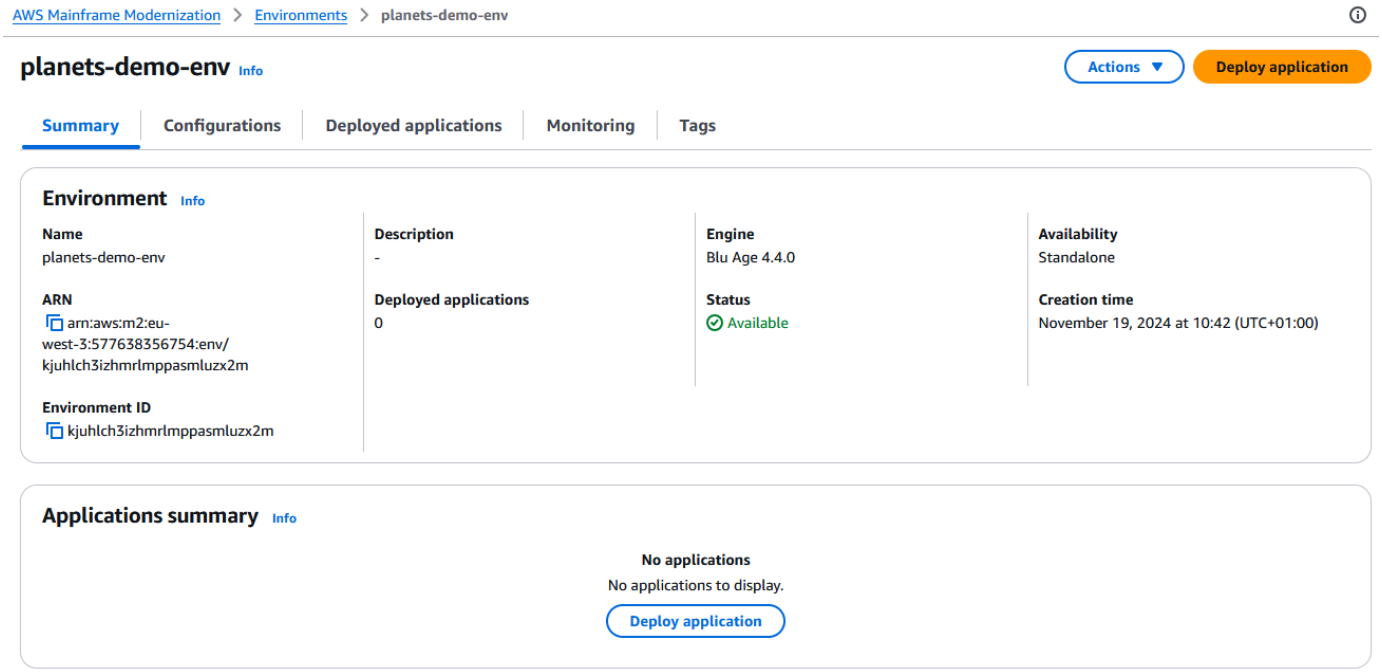
## Schritt 5: Stellen Sie eine Anwendung bereit

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:

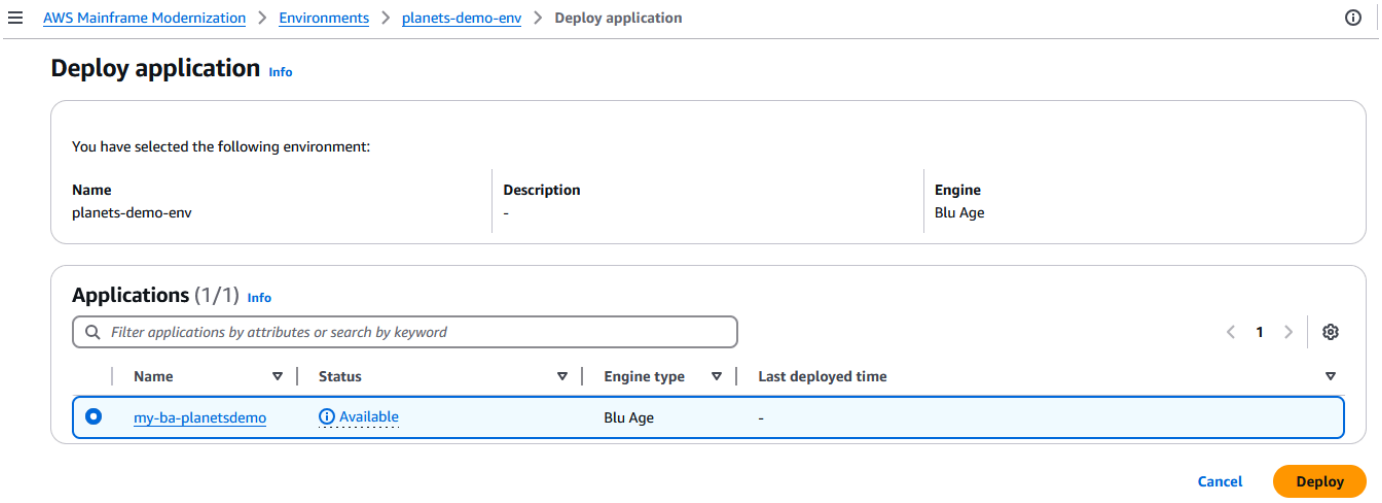
1. Navigieren Sie in der AWS Management Console zu AWS Mainframe Modernization. Wählen Sie im Navigationsbereich Environments (Umgebungen) aus. Die Seite mit der Liste der Umgebungen wird angezeigt.




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



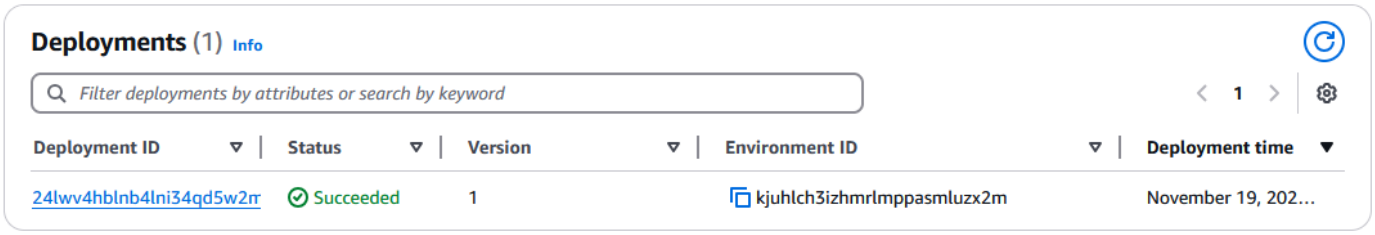

4. Wählen Sie die zuvor erstellte Anwendung und dann die Version aus, für die Sie Ihre Anwendung bereitstellen möchten. Wählen Sie dann Deploy (Bereitstellen) aus.




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.




3. Wählen Sie Aktionen und anschließend Anwendung starten.

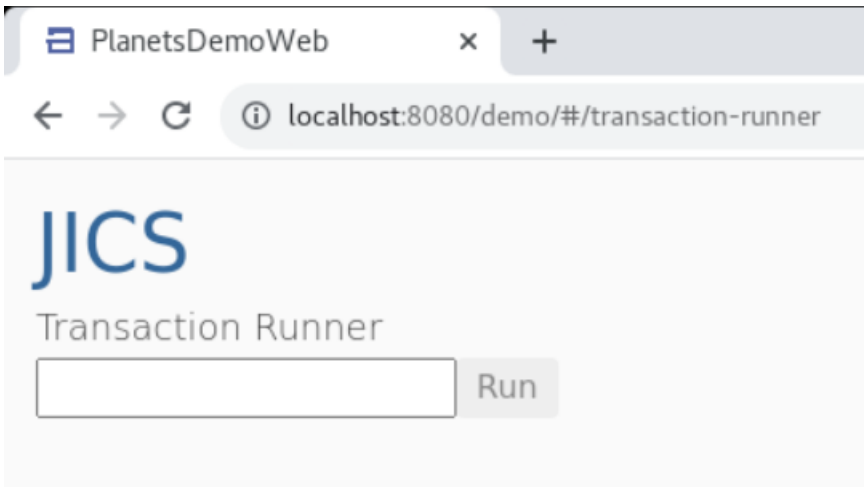
## Schritt 7: Greifen Sie auf die Anwendung zu

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 unterFehler zur Fehlerbehebung: Auf eine Anwendungs-URL kann nicht zugegriffen werden.

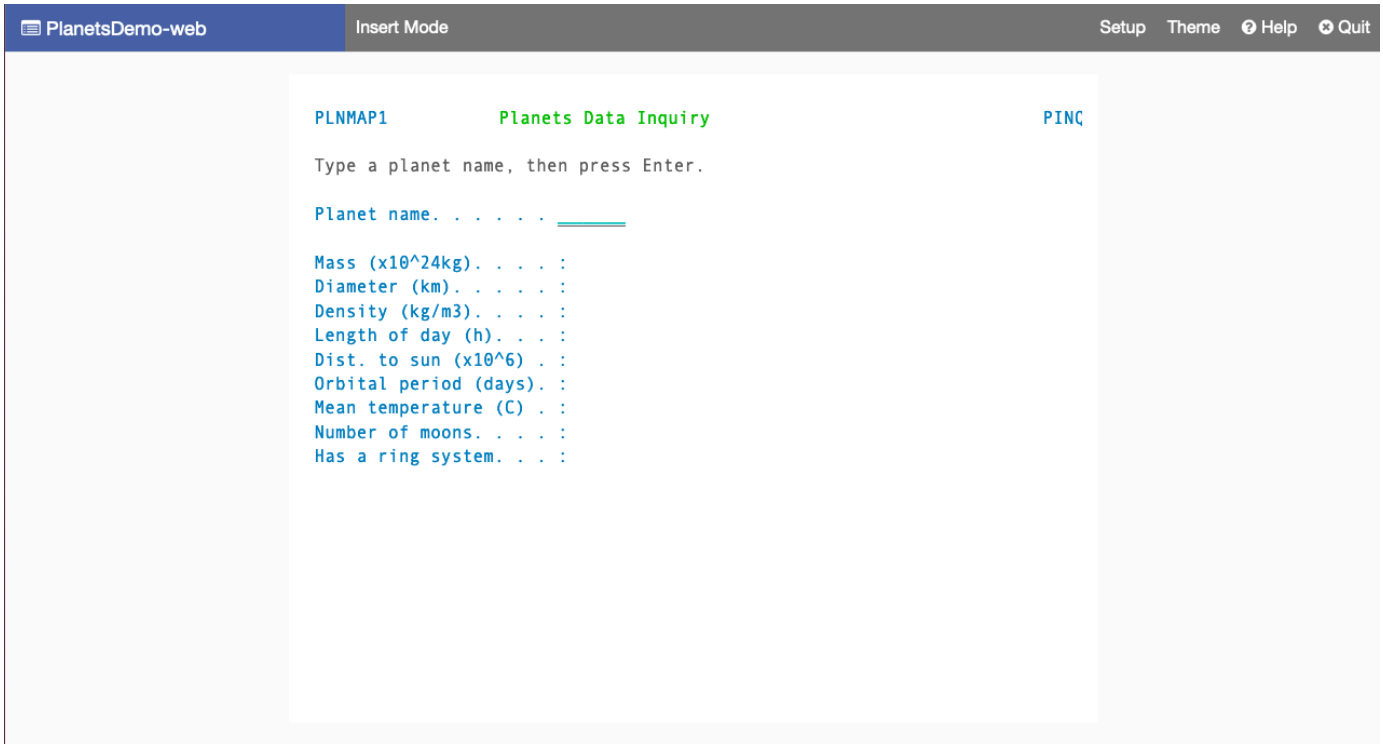
> **Note**
>
> Wenn auf die Anwendung nicht zugegriffen werden kann und in der Regel für eingehende Nachrichten für die Sicherheitsgruppe auf Port 8196 die Option „Meine IP“ ausgewählt ist, geben Sie die Regel an, um Datenverkehr von LB I/P auf Port 8196 zuzulassen.

## Schritt 8: Testen der Anwendung

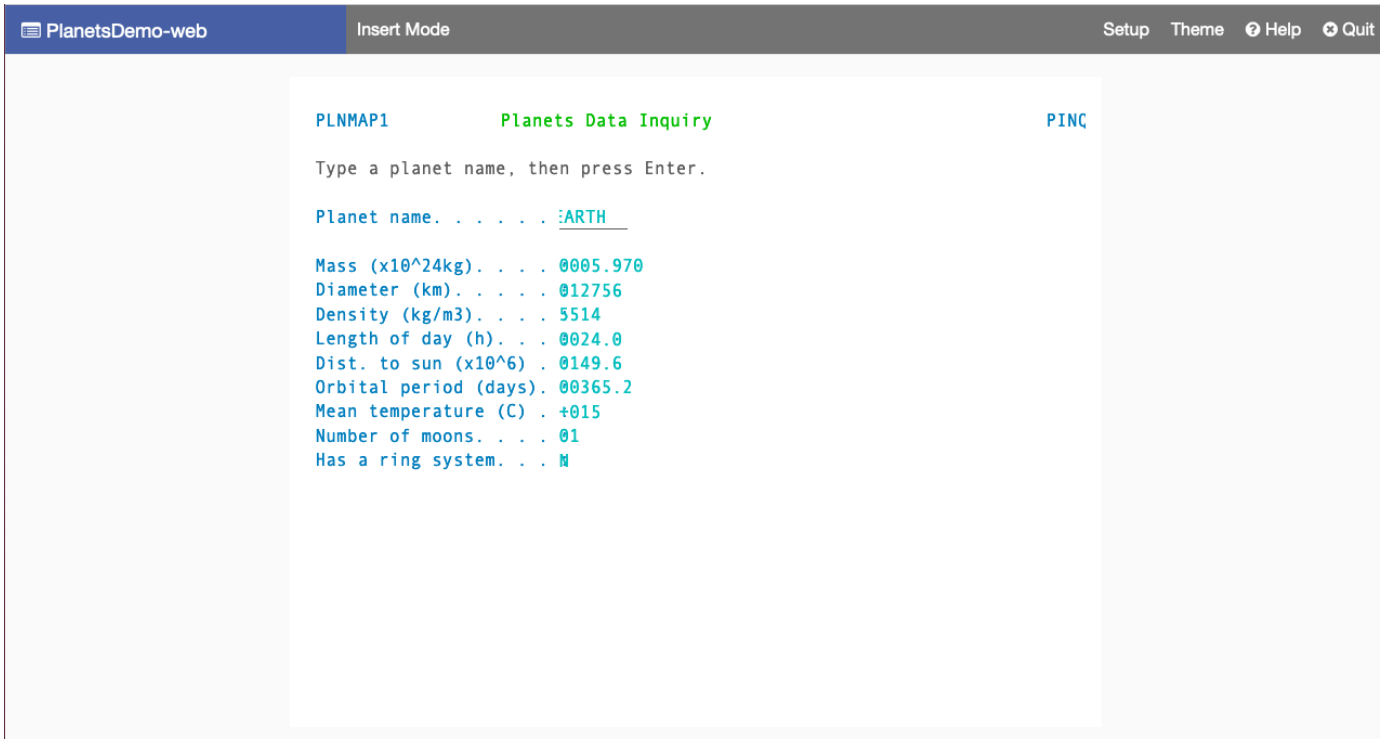
In diesem Schritt führen Sie eine Transaktion in der migrierten Anwendung aus.

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 Modernization Anwendung.
- Löschen Sie die Laufzeitumgebung. Weitere Informationen finden Sie unter Löschen Sie eine Laufzeitumgebung für die AWS Mainframe-Modernisierung.

# Tutorial: Verwaltete Laufzeit für Rocket Software (ehemals Micro Focus) einrichten

Mit der Runtime-Engine von Rocket Software können Sie eine Anwendung in einer von AWS Mainframe Modernization verwalteten Laufzeitumgebung bereitstellen und ausführen. Dieses Tutorial zeigt, wie Sie die CardDemo Beispielanwendung in einer von AWS Mainframe Modernization verwalteten Laufzeitumgebung mit der Runtime-Engine von Rocket Software 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 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: Fügen Sie dem Secret den SSLMode hinzu

- Schritt 6: Erstellen Sie eine Laufzeitumgebung
- Schritt 7: Erstellen Sie eine Anwendung
- Schritt 8: Stellen Sie eine Anwendung bereit
- Schritt 9: Datensätze importieren
- Schritt 10: Starten Sie eine Anwendung
- Schritt 11: 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 Rocket Software-Instanz starten und den Rumba 3270-Emulator verwenden (nicht kostenlos erhältlich).

  Informationen AppStream zu 2.0 finden Sie unter. the section called “Tutorial: AppStream 2.0 für Enterprise Analyzer und Enterprise Developer einrichten”

  > **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_runtime`
- 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

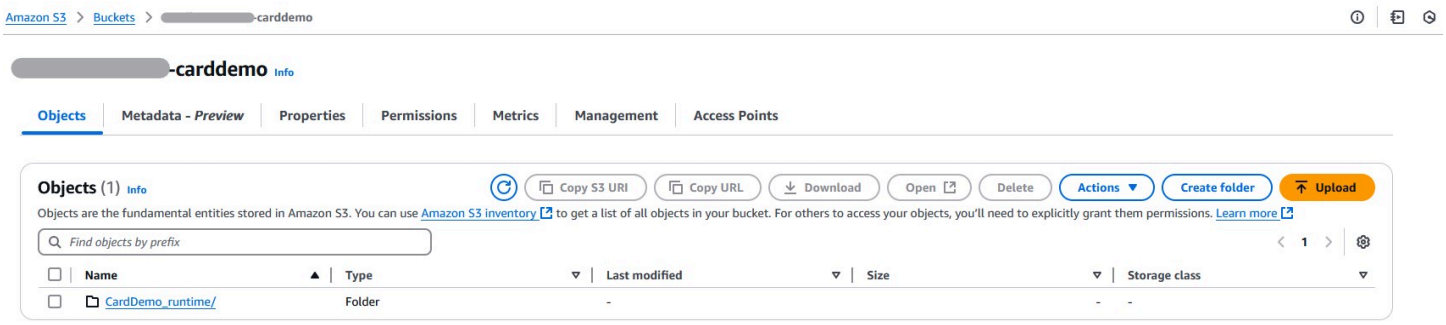
In diesem Schritt erstellen Sie einen Amazon S3 S3-Bucket und laden CardDemo Dateien in diesen Bucket hoch. Später in diesem Tutorial verwenden Sie diese Dateien, um die CardDemo Beispielanwendung in einer Managed Runtime-Umgebung von AWS Mainframe Modernization Rocket Software bereitzustellen und auszuführen.

> **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 Rocket Software 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_runtime` 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_runtime` Ordner.




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 aus.

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 markieren.
8. Wählen Sie Weiter aus.
9. Weisen Sie die Schlüsselverwendungsberechtigung zu, indem Sie das Kästchen neben Ihrem Benutzer oder Ihrer Rolle ankreuzen.
10. Wählen Sie Weiter aus.
11. Bearbeiten Sie auf dem Überprüfungsbildschirm die Schlüsselrichtlinie und geben Sie dann Folgendes in das bestehende Feld „Statement“ 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.
12. 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" : "*"
        }
    ]
}
```

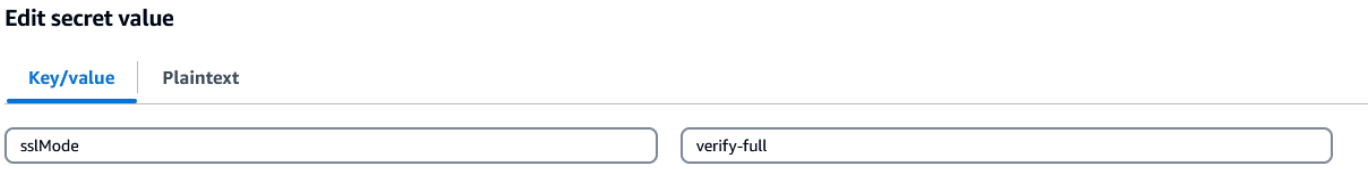
10. Wählen Sie Save aus.
11. Wählen Sie „Weiter“ für die nachfolgenden Bildschirme und anschließend „Speichern“.

## Schritt 5: Fügen Sie dem Secret den SSLMode hinzu

Um den SSLMode zum Secret hinzuzufügen

1. Aktualisieren Sie die Geheimnisliste, um das neue Geheimnis zu sehen.
2. Wählen Sie in Schritt 4 das neu erstellte Geheimnis aus und notieren Sie sich das, `Secret ARN` weil Sie es später im Tutorial benötigen.

3. Wählen Sie auf der Registerkarte „Übersicht“ des Geheimnisses die Option Geheimen Wert abrufen aus.
4. Wählen Sie „Bearbeiten“ und anschließend „Zeile hinzufügen“.
5. Fügen Sie einen Schlüssel für `sslMode` mit einem Wert von hinzu`verify-full`:

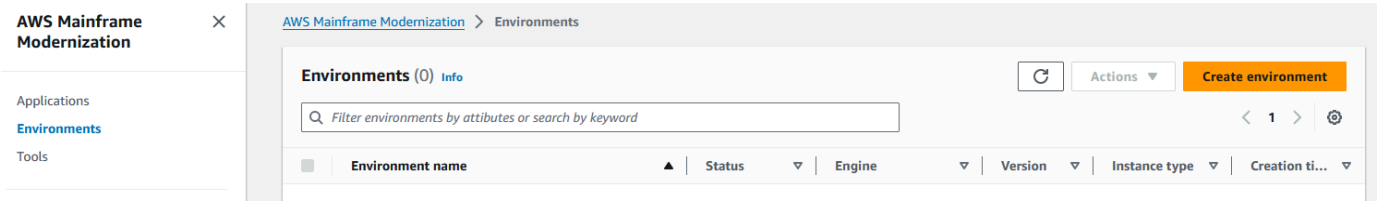



6. Wählen Sie Save aus.

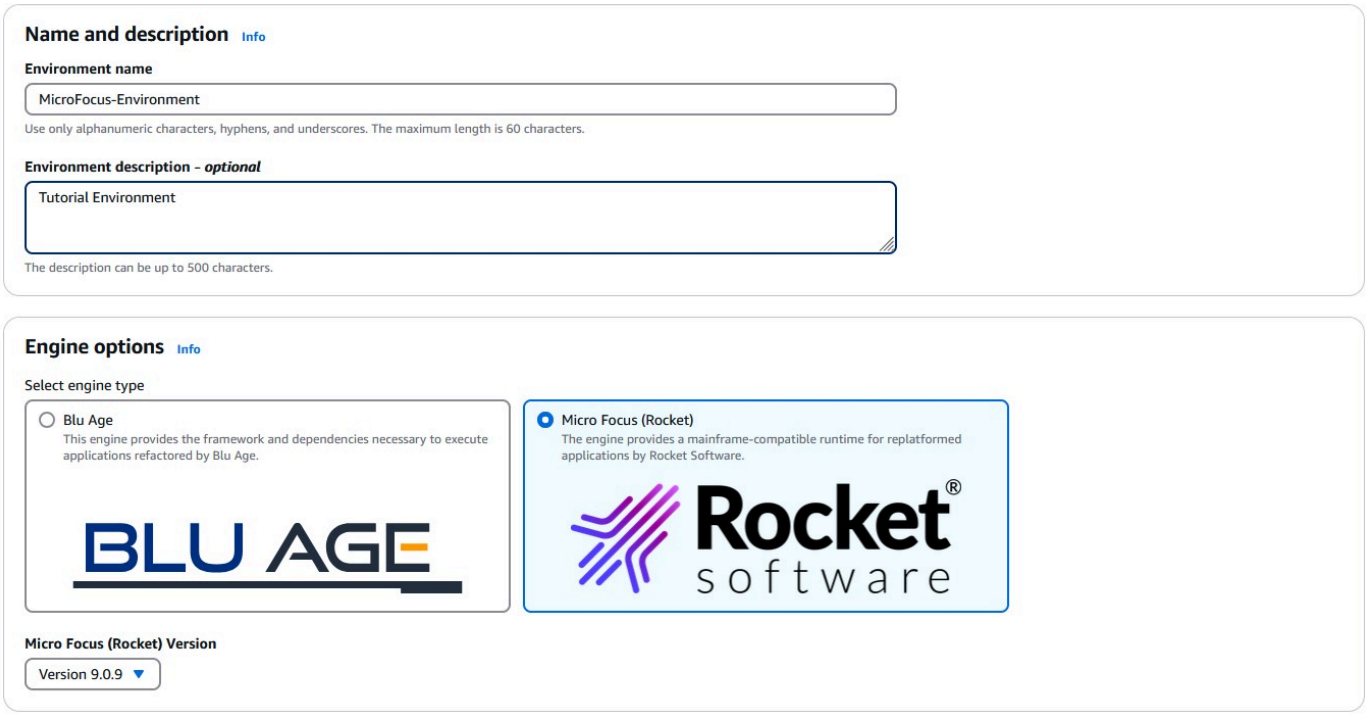
## Schritt 6: Erstellen Sie eine Laufzeitumgebung

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 (Rocket) ausgewählt ist.
   c. Wählen Sie die neueste Micro Focus (Rocket) -Version.
   d. Wählen Sie Weiter aus.




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 Instance-Typ und die Anzahl der gewünschten Instances 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

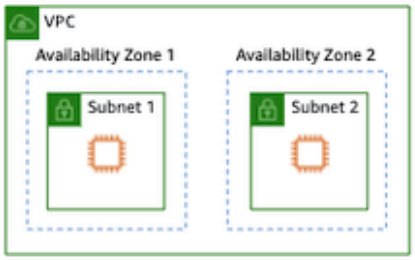
## Availability Info

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.

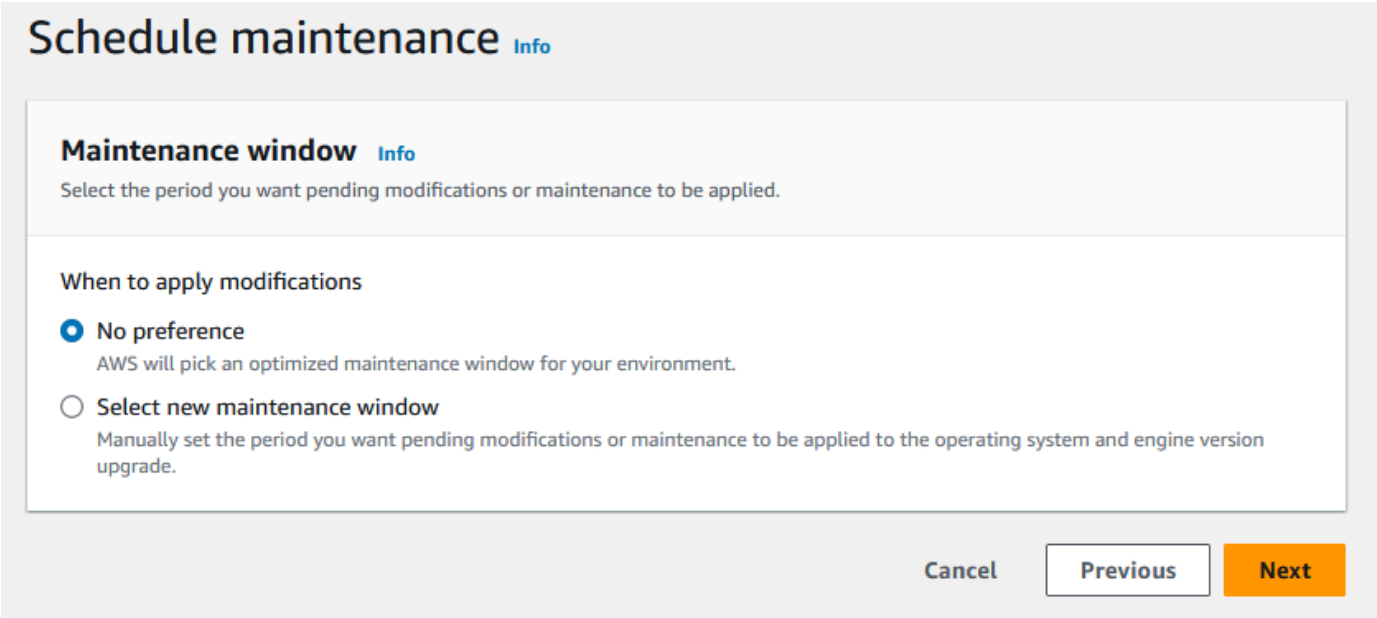
Choose subnets

subnet-56f1a | us-west-2a ×

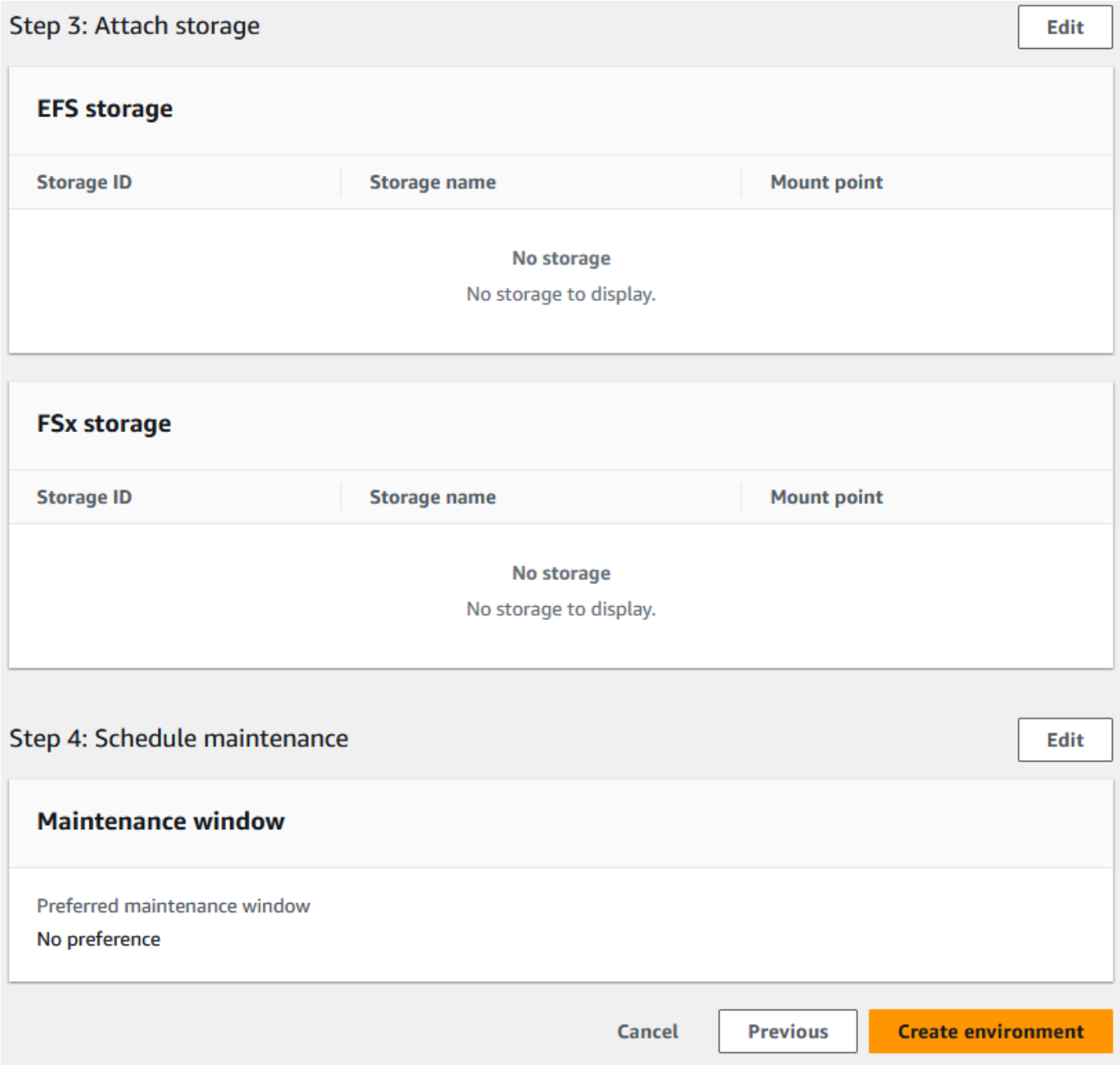
subnet-6685 | us-west-2b ×

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

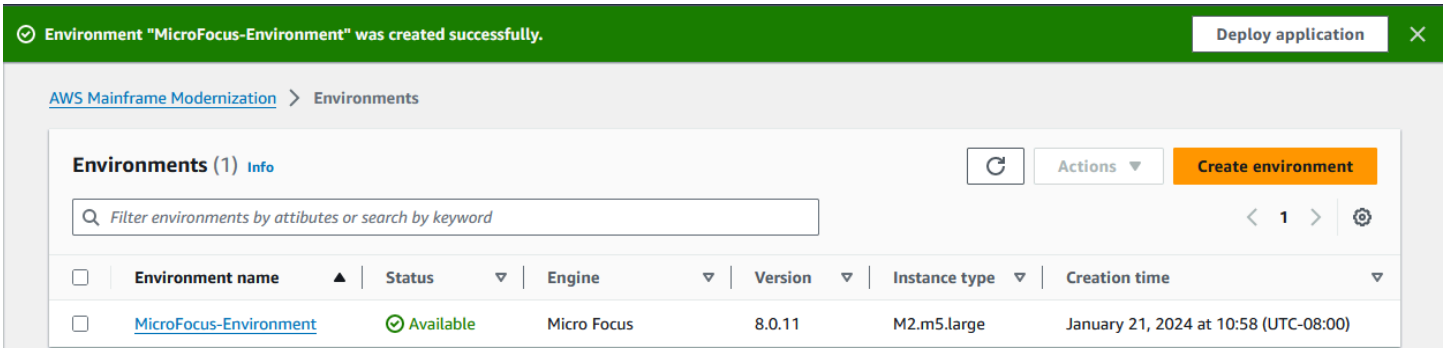
5. Wählen Sie auf der Seite Speicher anhängen die Option Weiter aus.
6. Wählen Sie auf der Seite Wartung planen die Option Keine Präferenz und dann Weiter aus.




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.



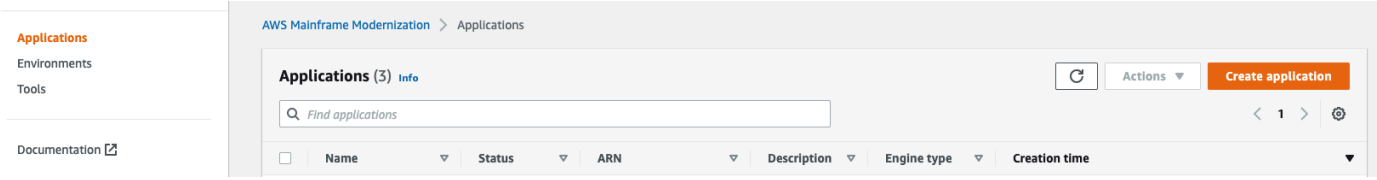

Wenn Sie Ihre Umgebung erstellt haben, erscheint ein Banner mit der Aufschrift`Environment` *name* `was created successfully`, und das Feld Status ändert sich zu Verfügbar. Die Erstellung der Umgebung dauert einige Minuten, aber Sie können mit den nächsten Schritten fortfahren, während sie ausgeführt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



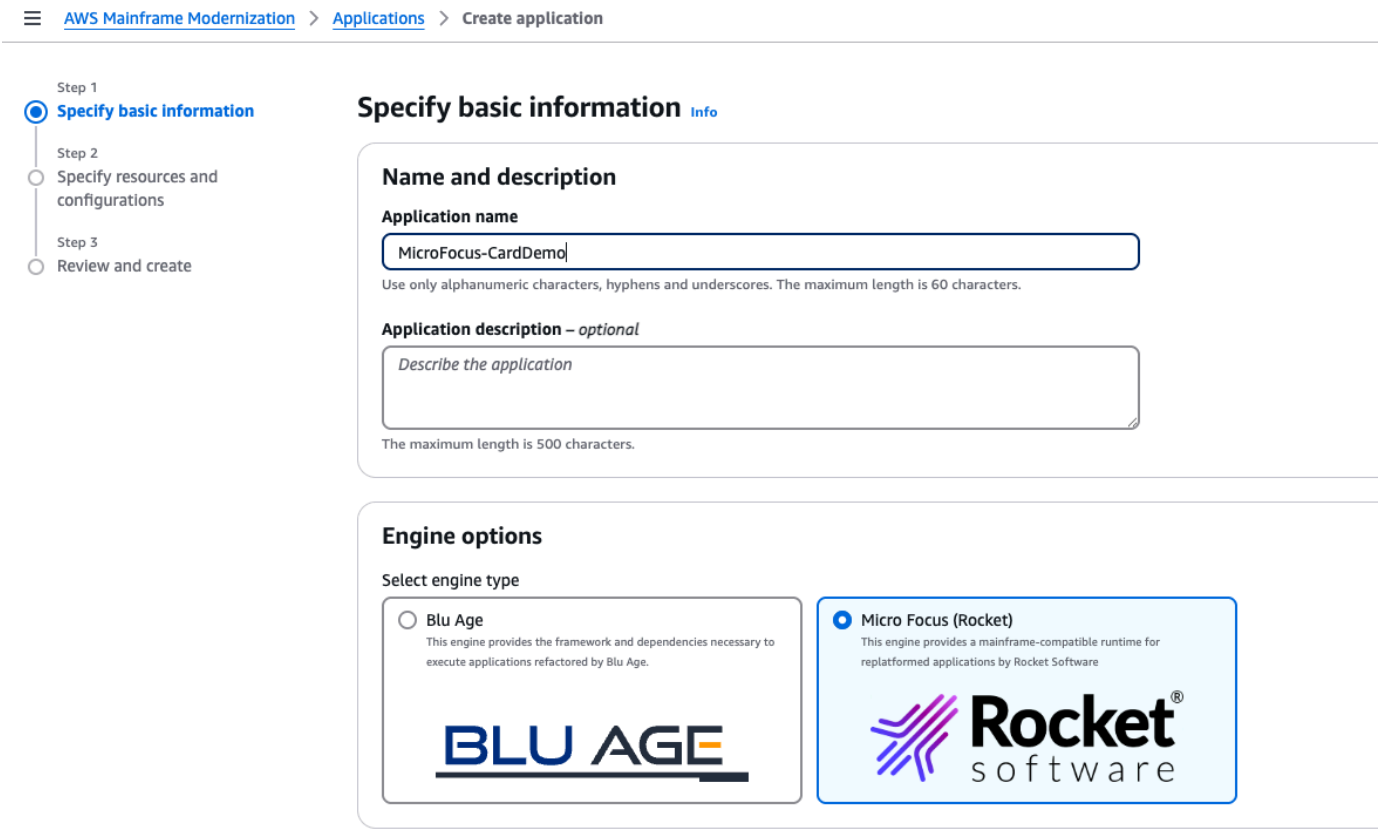

# Schritt 7: Erstellen Sie eine Anwendung

So erstellen Sie eine Anwendung

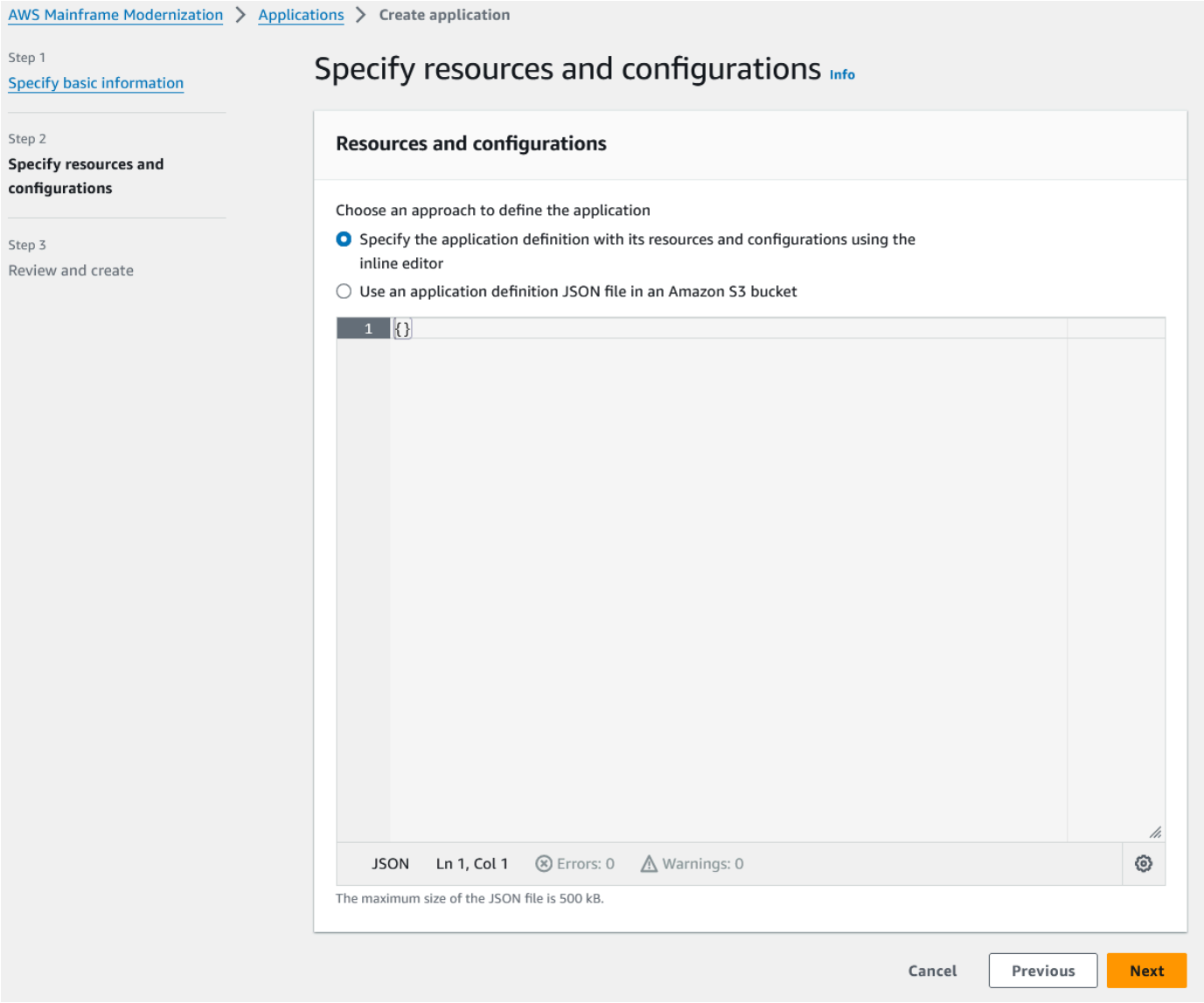
1. Wählen Sie im Navigationsbereich Applications (Anwendungen). Wählen Sie dann Anwendung erstellen.




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 (Rocket) ausgewählt ist. Wählen Sie anschließend Weiter.




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




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_runtime"
      }
    }
```

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

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

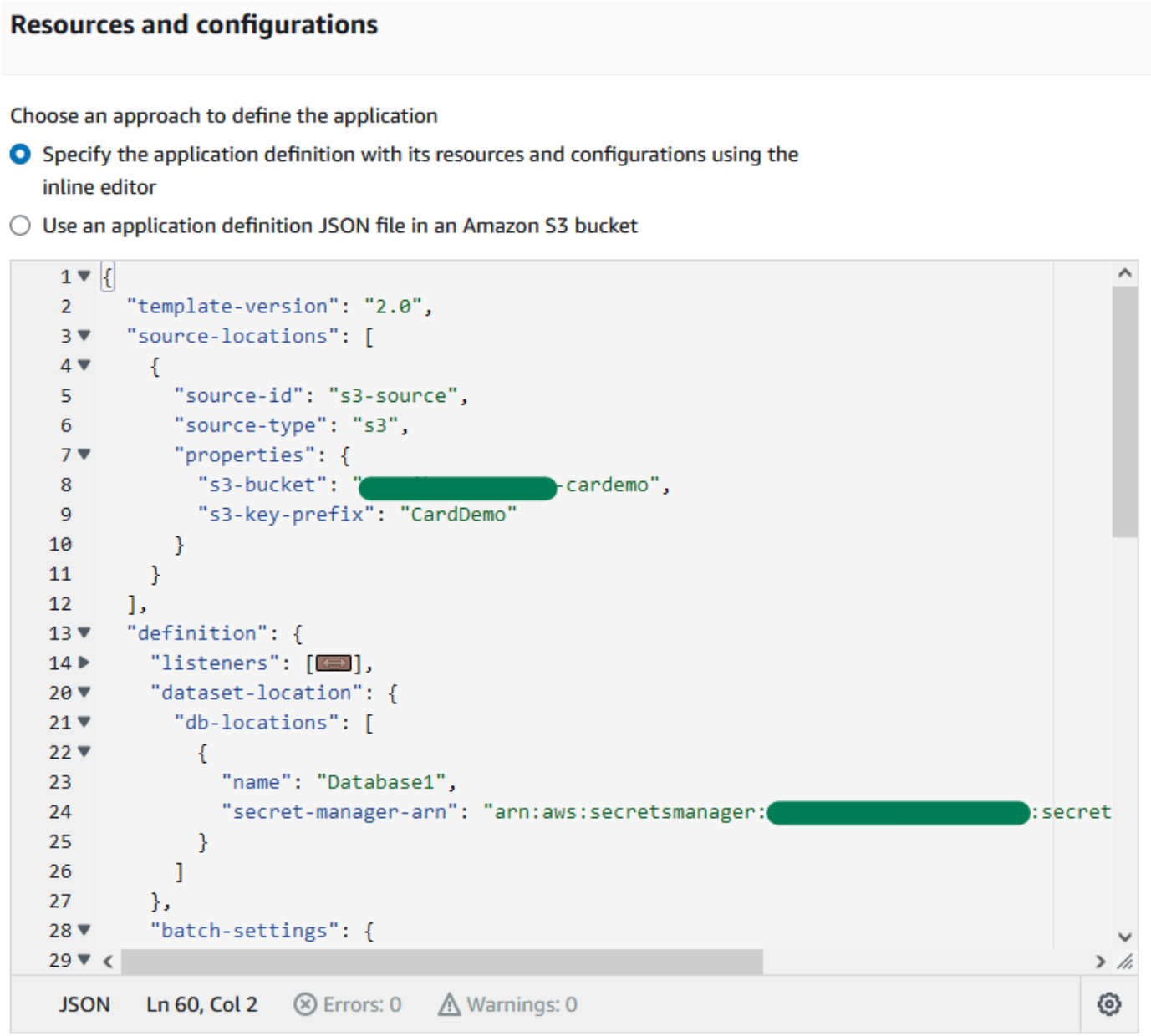
> **Note**
>
> Diese Datei kann sich ändern.

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.

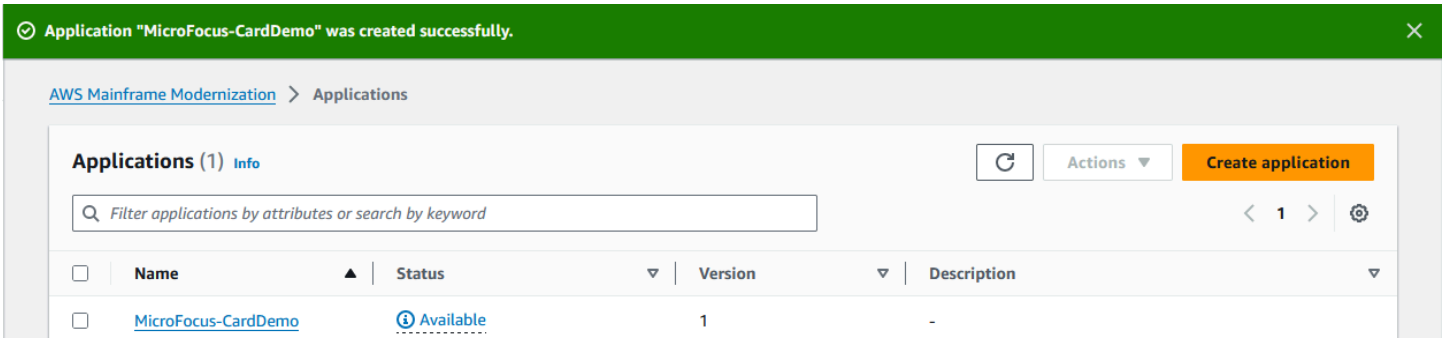
   b. Ersetzen Sie den Wert für `s3-key-prefix` durch den Ordner (key prefix), in den Sie die CardDemo Beispieldateien hochgeladen haben. Wenn Sie das `CardDemo` Verzeichnis direkt in einen Amazon S3 S3-Bucket hochgeladen haben, muss `s3-key-prefix` das nicht geändert werden.

   c. Ersetzen Sie beide `secret-manager-arn` Werte durch den ARN für das Datenbankgeheimnis, das Sie in Schritt 4 erstellt haben.




Weitere Informationen zur Anwendungsdefinition finden Sie unterAnwendungsdefinition von Rocket Software (ehemals Micro Focus).

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.

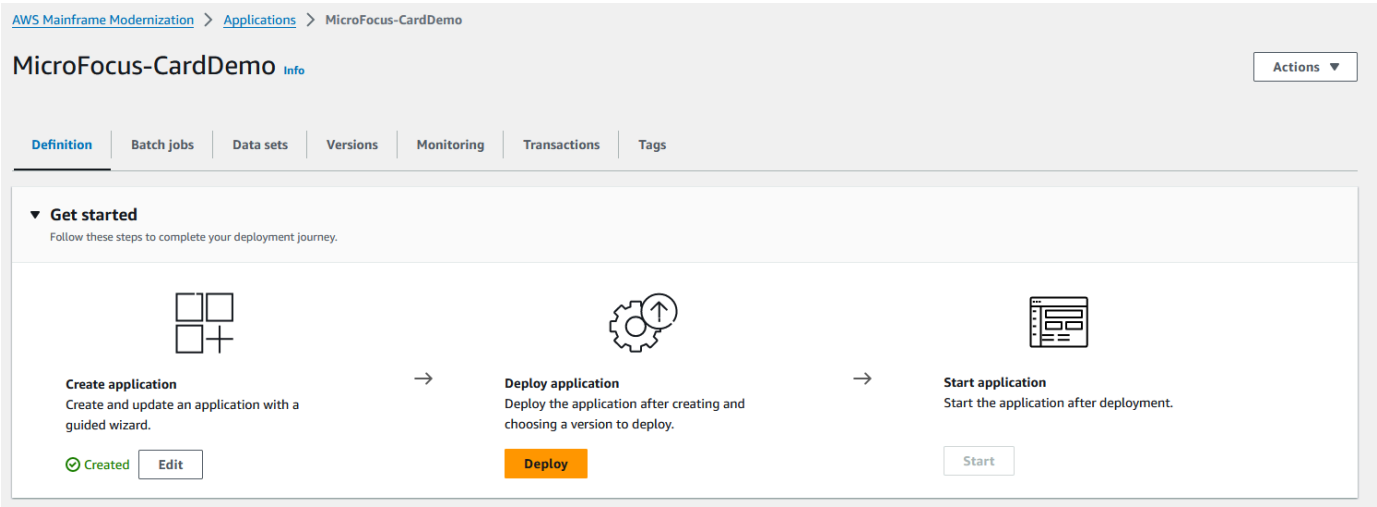



Wenn Sie Ihre Anwendung erstellt haben, erscheint ein Banner mit der Aufschrift`Application` *`name`* `was created successfully`. Und das Statusfeld ändert sich zu Verfügbar.

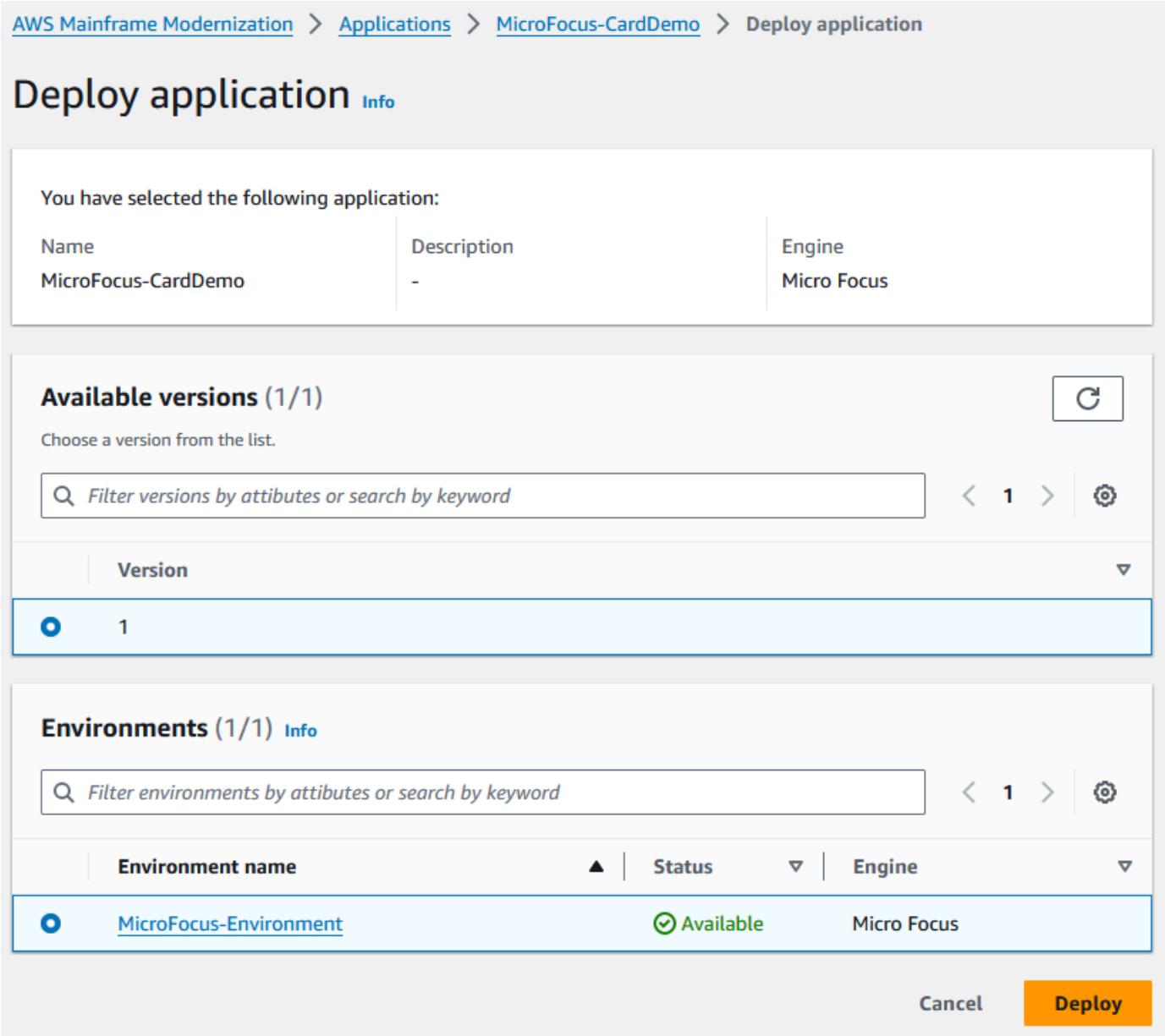
## Schritt 8: Stellen Sie eine Anwendung bereit

Um eine Anwendung bereitzustellen

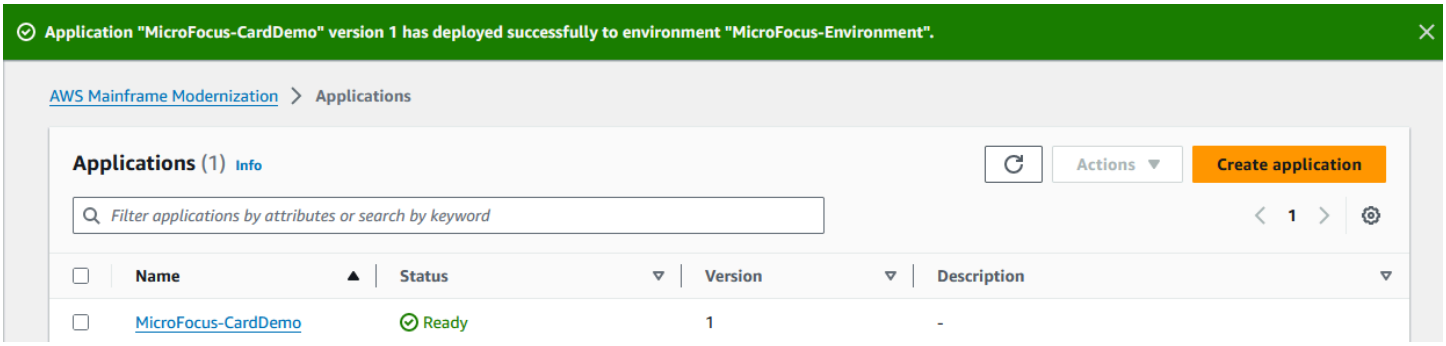
1. Wählen Sie im Navigationsbereich Anwendungen und dann aus`MicroFocus-CardDemo`.
2. Wählen Sie unter Anwendung bereitstellen die Option Bereitstellen aus.




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



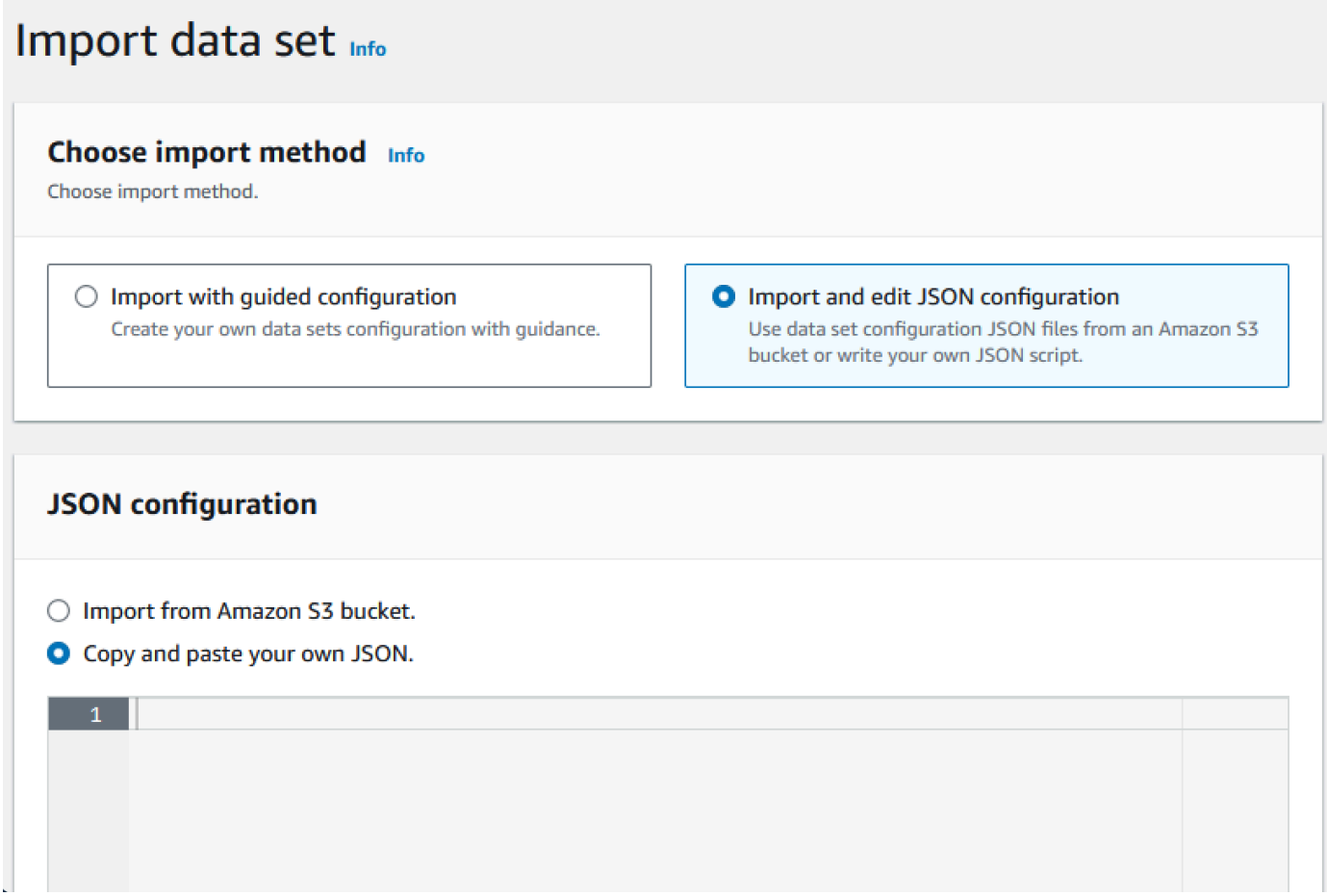

Wenn die CardDemo Anwendung erfolgreich bereitgestellt wurde, ändert sich der Status in Bereit.



# Schritt 9: 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.




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_runtime/catalog/data/
AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS.DAT"
            }
        },
        {
            "dataSet": {
                "storageType": "Database",
                "datasetName": "AWS.M2.CARDDEMO.CARDDATA.VSAM.AIX.PATH",
                "relativePath": "DATA",
                "datasetOrg": {
                    "vsam": {
                        "format": "KS",
                        "encoding": "A",
                        "primaryKey": {
                            "length": 11,
                            "offset": 16
                        }
                    }
                },
                "recordLength": {
                    "min": 150,
                    "max": 150
                }
            },
            "externalLocation": {
```

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

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

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

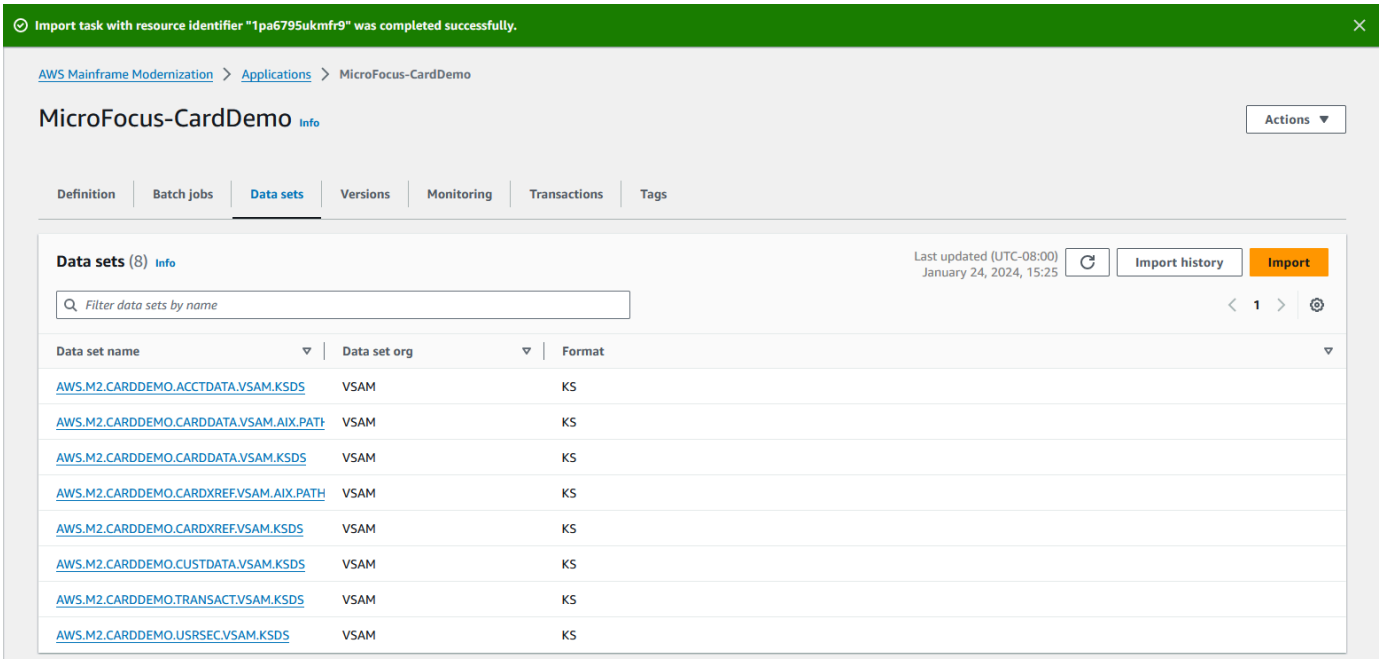
```
                "datasetName": "AWS.M2.CARDDEMO.USRSEC.VSAM.KSDS",
                "relativePath": "DATA",
                "datasetOrg": {
                    "vsam": {
                        "format": "KS",
                        "encoding": "A",
                        "primaryKey": {
      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

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

> **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.

6. 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.

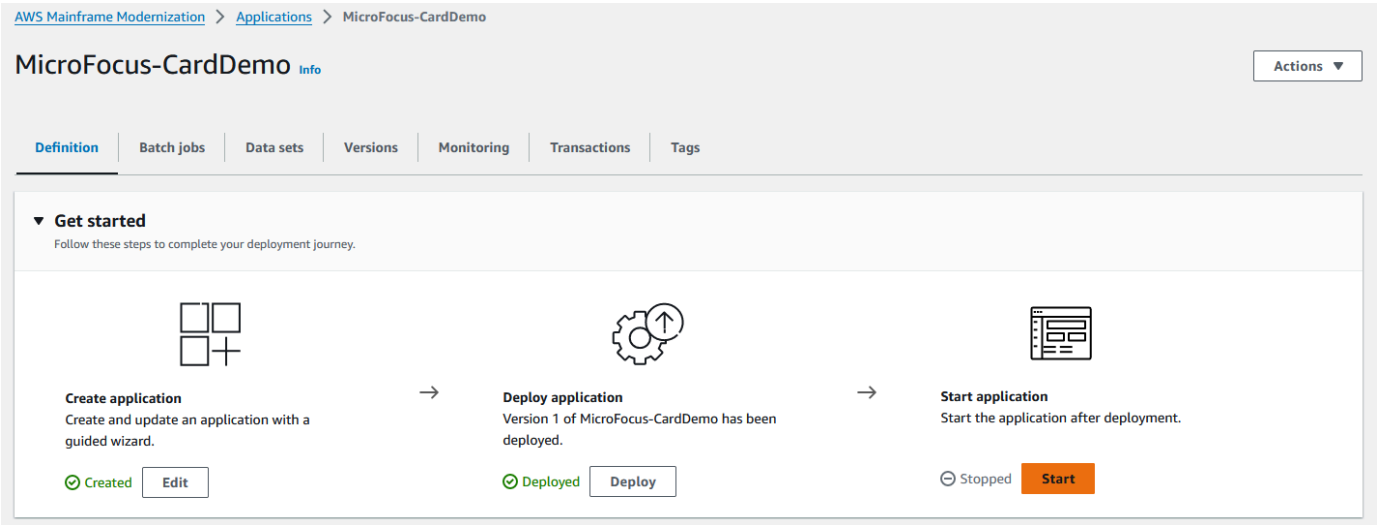



Sie können auch den Status aller Datensatz-Importe anzeigen, indem Sie auf der Registerkarte Datensätze die Option Importverlauf auswählen.

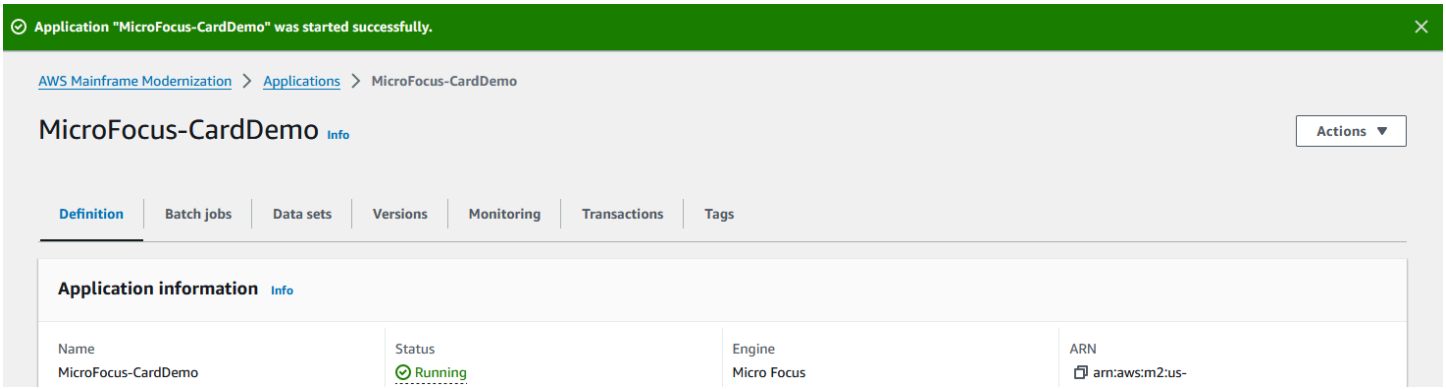
## Schritt 10: Starten Sie eine Anwendung

Um eine Anwendung zu starten

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



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 11: 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.

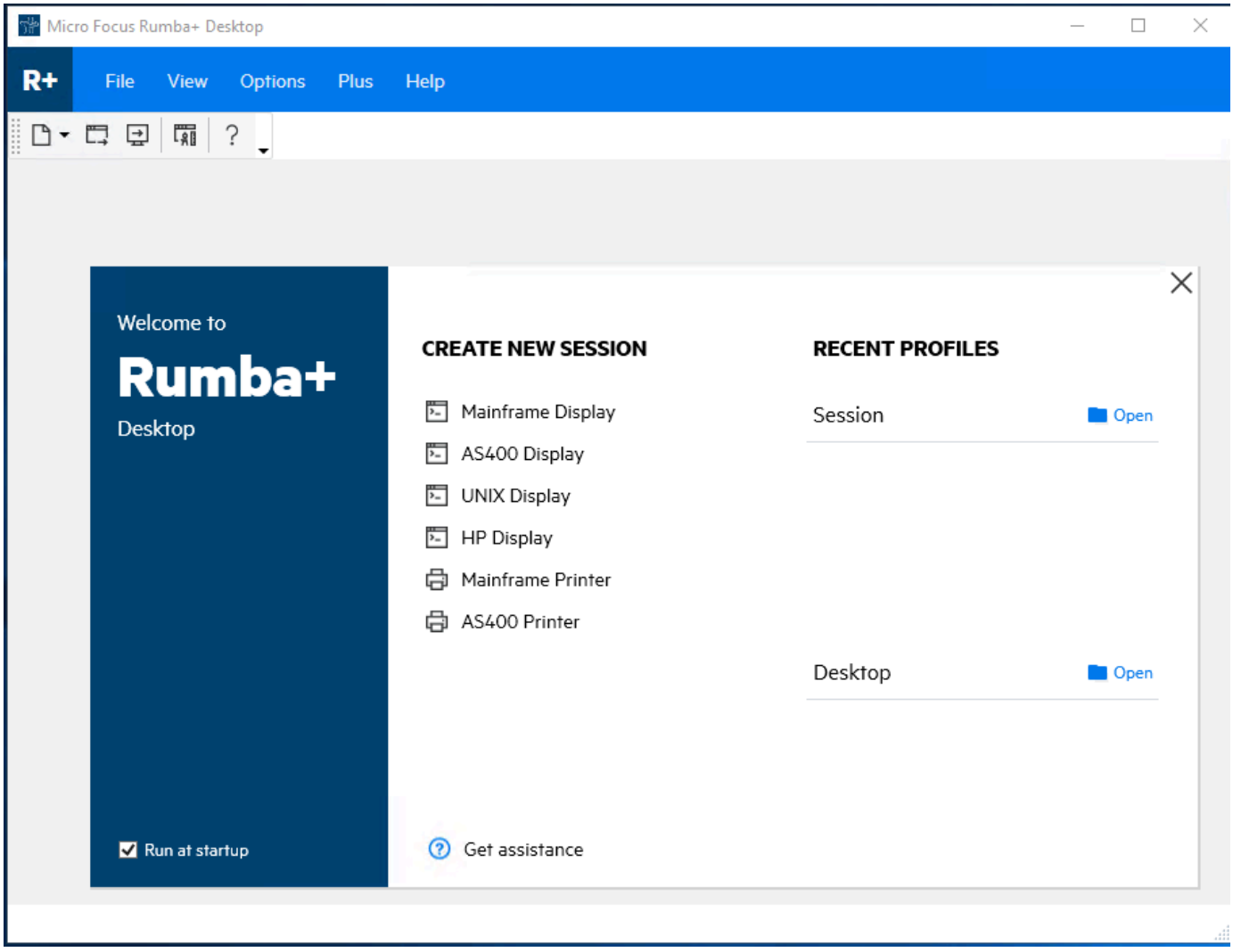
Um die TN327 0-Verbindung zu konfigurieren, benötigen Sie außerdem den DNS-Hostnamen und den Port der Anwendung.

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

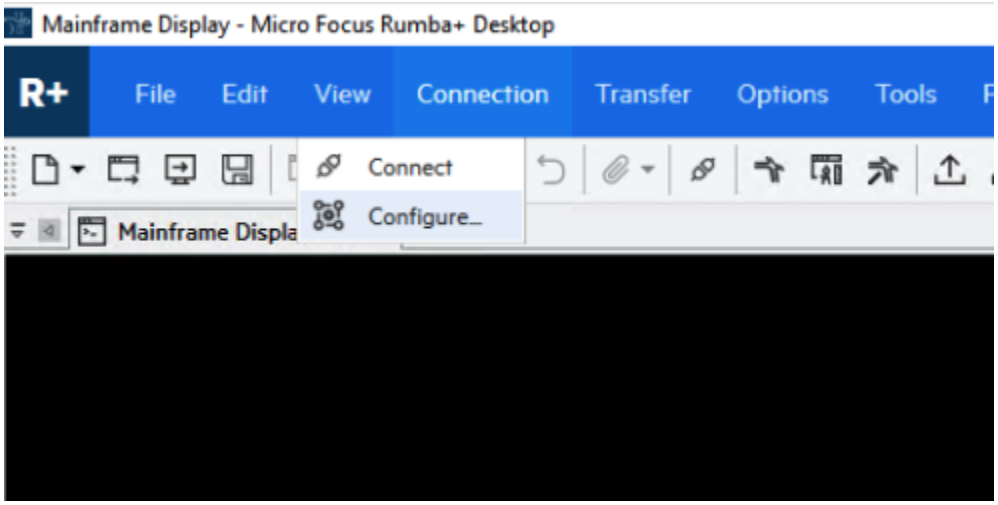
1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole und wählen Sie Applications und anschließend. `MicroFocus-CardDemo`
2. Wählen Sie das Kopiersymbol, um den DNS-Hostnamen zu kopieren. Notieren Sie sich auch die Port-Nummer.
3. Starten Sie einen Terminalemulator. In diesem Tutorial wird Micro Focus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

   > **Note**
   >
   > Die Konfigurationsschritte variieren je nach Emulator.

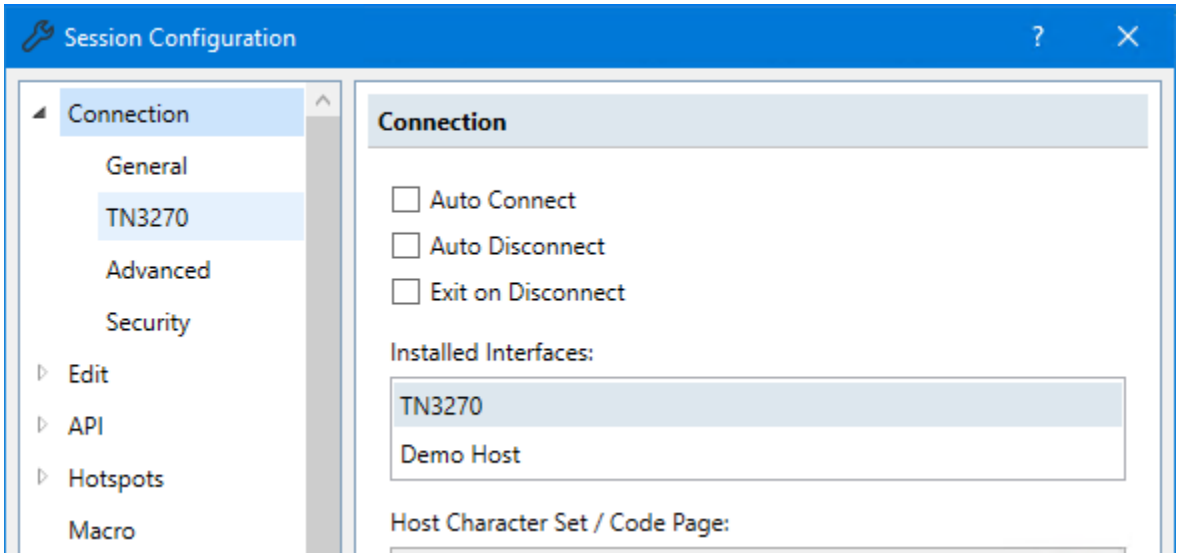
4. Wählen Sie Mainframe Display.



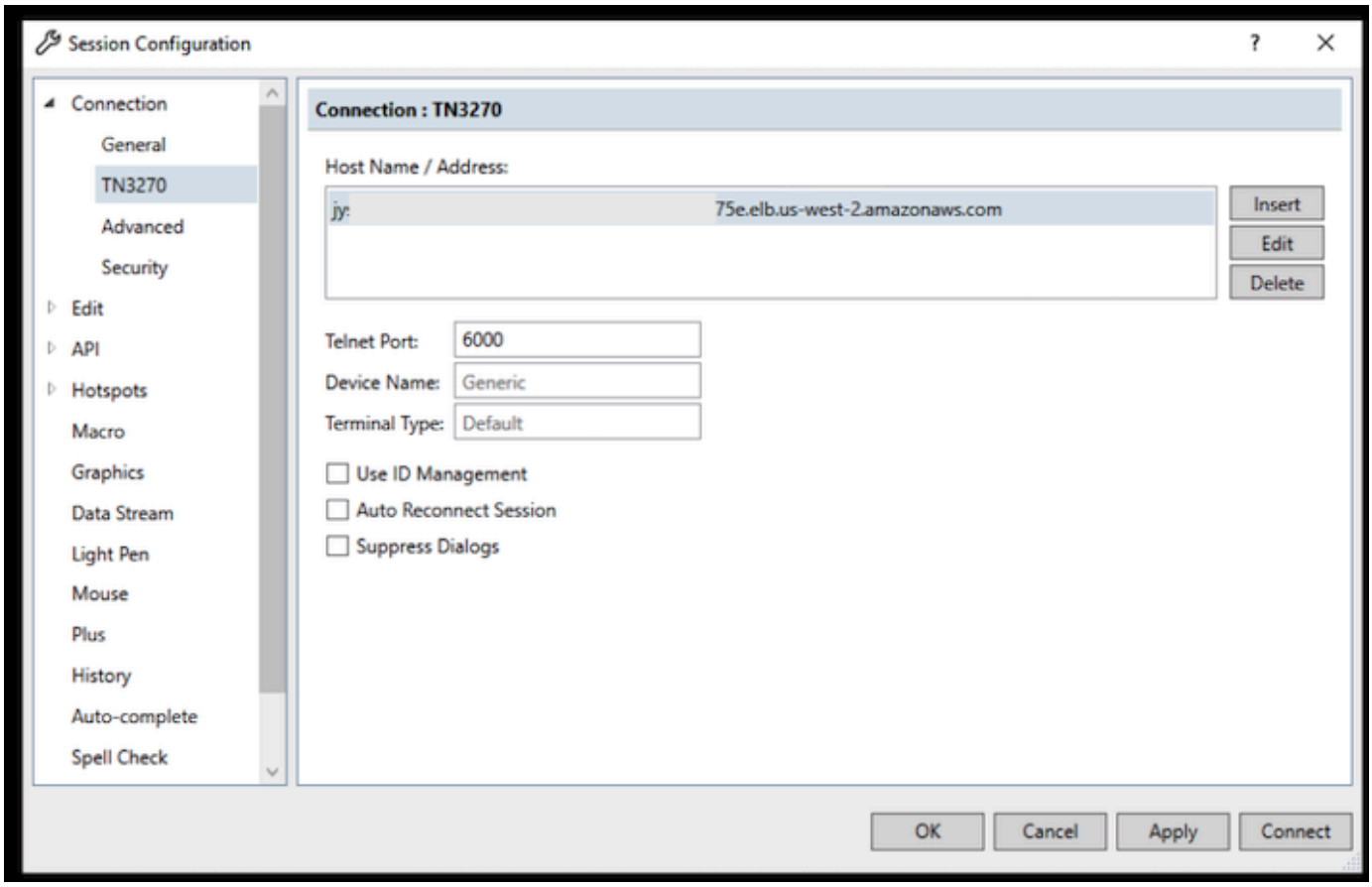

5. Wählen Sie „Verbindung“ und anschließend „Konfigurieren“.




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



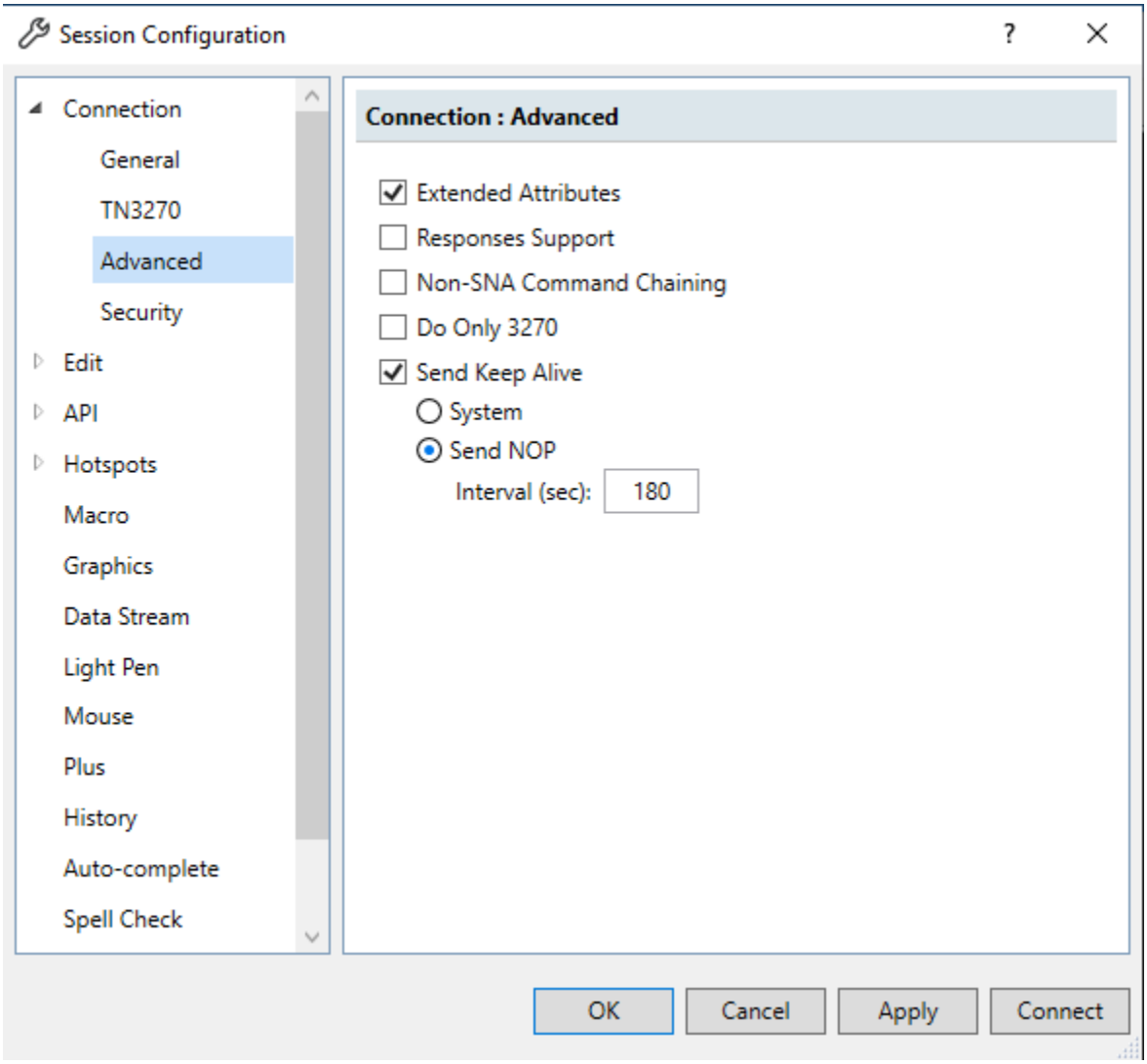

7. Wählen Sie Einfügen und fügen Sie das DNS `Hostname` für die Anwendung ein. Geben Sie `6000` für den Telnet-Port an.



> **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 TN327 0-Terminal auf mindestens 180 Sekunden konfigurieren, können Sie sicherstellen, dass der Network Load Balancer Ihre Verbindung nicht unterbricht.



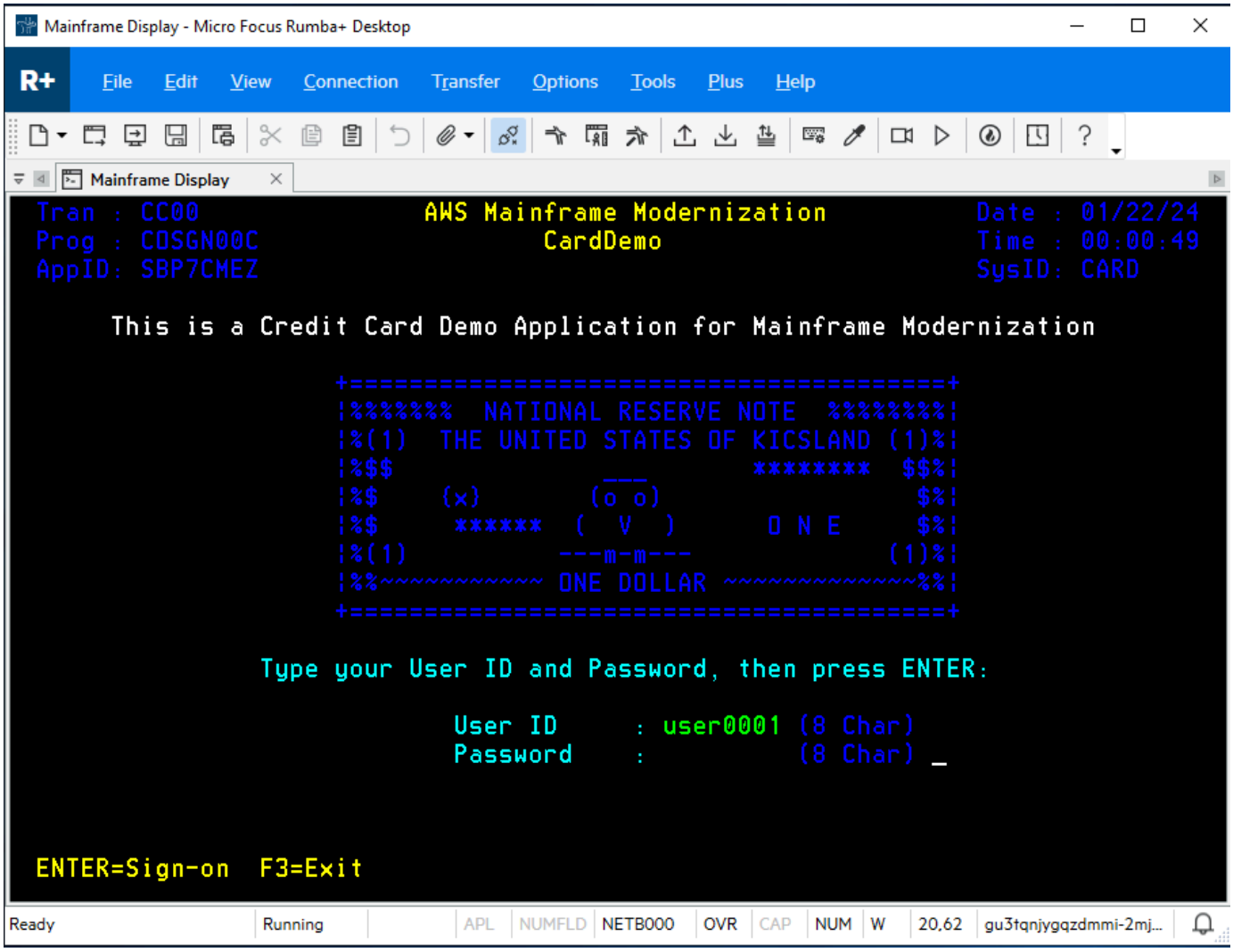

9. Wählen Sie Connect aus.

> **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.

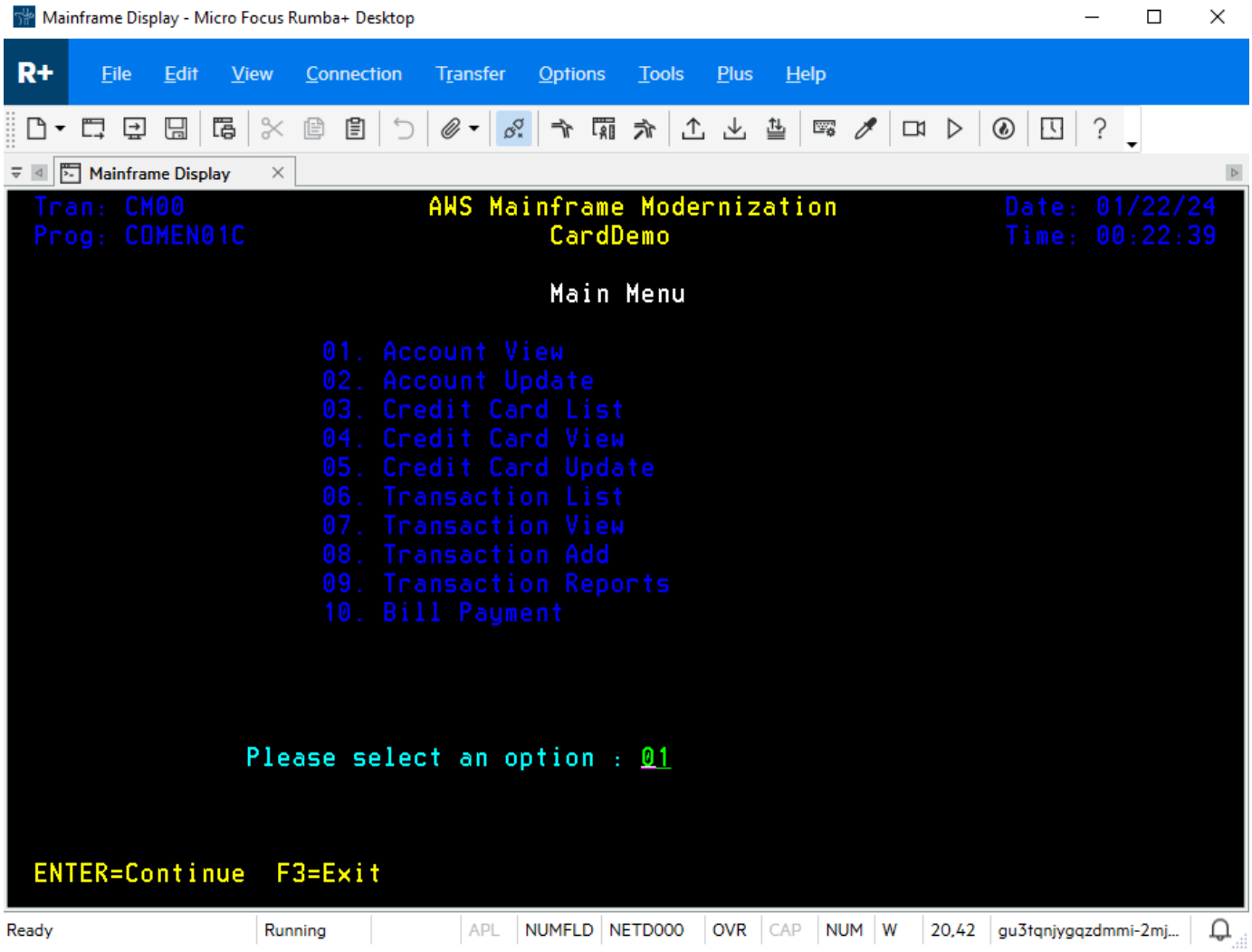
- Versuchen Sie als Diagnoseschritt, die Sicherheitsgruppenregeln für den eingehenden Datenverkehr für das Programm hinzuzufügen oder zu ändern, um Datenverkehr über Port 6000 von überall her zuzulassen (d. h. CIDR Block 0.0.0.0/0). Wenn Sie erfolgreich eine Verbindung herstellen, wissen Sie, dass die 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.

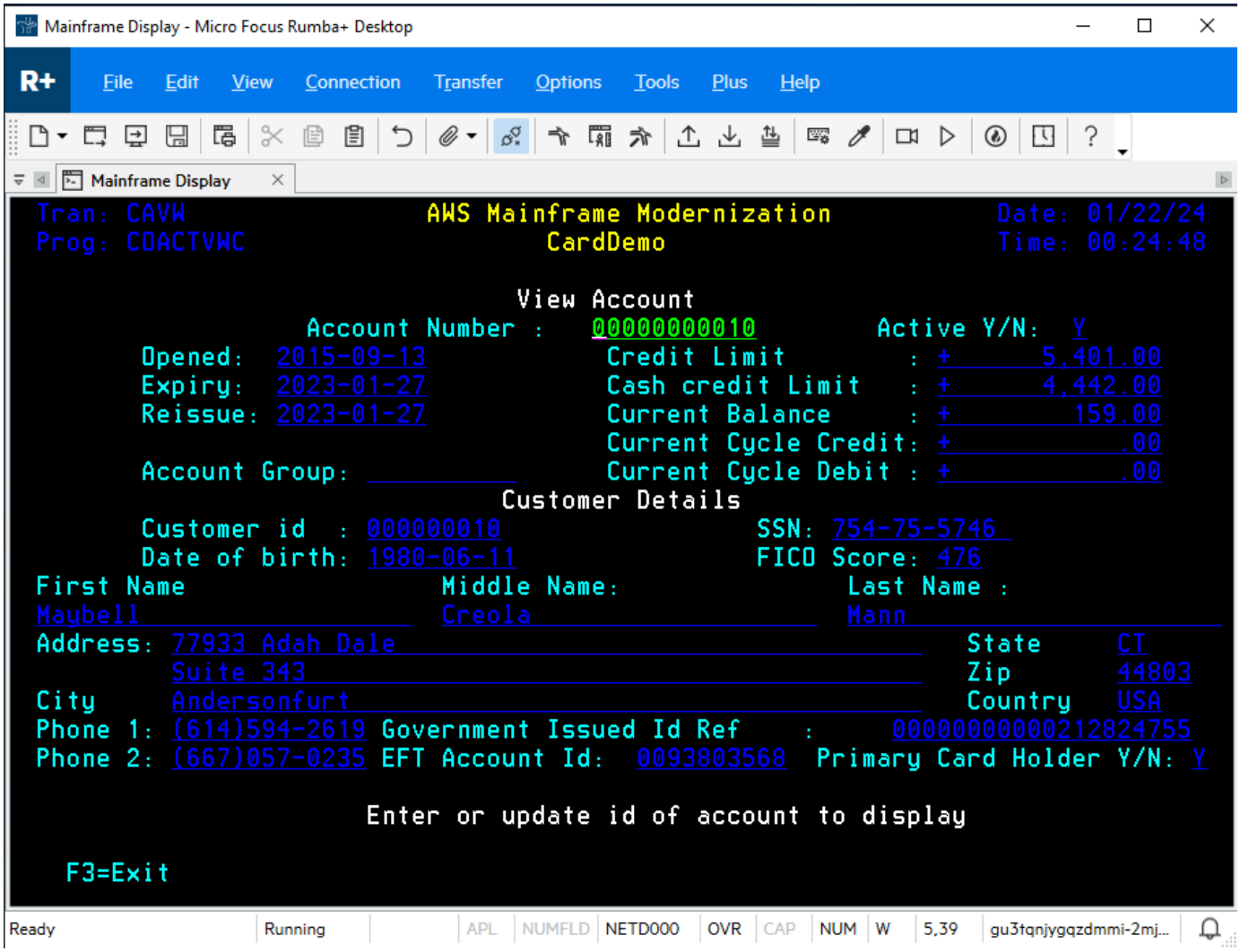



11. Nachdem Sie sich erfolgreich angemeldet haben, können Sie durch die CardDemo Anwendung navigieren.
12. Geben Sie `01` für die Kontoansicht ein.




13. Geben Sie `00000000010` die Kontonummer ein und drücken Sie die Eingabetaste auf Ihrer Tastatur.

> **Note**
>
> Andere gültige Konten sind `0000000011` und`00000000020`.




14. Drücken SieF3, 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 Modernization Anwendung.
- 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

Wie Sie eine Entwicklungsumgebung für Ihre modernisierten Anwendungen einrichten, erfahren Sie unter Tutorial: Setup AppStream 2.0 für die Verwendung mit Rocket Enterprise Analyzer und Rocket Enterprise Developer.

# AWS Lebenszyklus von Mainframe-Modernisierungskomponenten

Jede Komponente der AWS Mainframe-Modernisierung durchläuft Versionsupgrades und einen Entwicklungszyklus. Auf dieser Seite können Sie sich einen Überblick über diese Komponenten, ihre Versions-Upgrade-Pläne und die Art und Weise verschaffen, wie AWS Mainframe Modernization die Veröffentlichung oder das Verfallsdatum dieser Komponenten oder ihrer Versionen kommuniziert.

## Überblick über den Lebenszyklus der Komponenten

AWS Der Lebenszyklus der Mainframe-Modernisierung beschreibt den Ansatz und die Zeitpläne für die Veröffentlichung und Unterstützung der Servicekomponenten für die AWS Mainframe-Modernisierung während ihres gesamten Lebenszyklus. Die Bereitstellung eines vorhersehbaren und konsistenten Lebenszyklus hilft Ihnen bei der Planung, dem Testen und der Bereitstellung neuerer Versionen.

Alle AWS bereitgestellten AWS Mainframe-Modernisierungskomponenten profitieren vom Produktsupport, der Support vom Zeitpunkt ihrer Veröffentlichung bis zu ihrer Außerbetriebnahme gemäß dem Veröffentlichungskalender der einzelnen Komponenten bereitgestellt wird. Weitere Informationen zum Support Umfang und zu den Aktivitäten finden Sie unter Tarife vergleichen Support. Bei aktiven Modernisierungsprojekten empfehlen wir in der Regel, dass der Kundensupport zunächst von professionellen Serviceteams gemäß Leistungsbeschreibung erbracht wird.

AWS Mainframe Modernization veröffentlicht einige Komponenten mit Versionen, die von Lieferanten stammen, bei denen es sich um das Unternehmen AWS selbst, ausgewählte AWS Partner oder Communitys handeln kann. Für jede AWS Mainframe-Modernisierungskomponente hat eine Version eine Hauptversionsnummer und eine Nebenversionsnummer. Jede Komponente hat ihre eigene Haupt- und Nebenversionsnummerierung.

Für versionierte Komponenten haben wir die folgenden Absichten:

- Wir wollen regelmäßig oder auf Kundenwunsch neuere Versionen von AWS Mainframe-Modernisierungskomponenten veröffentlichen. Wenn die neuere Version einer Komponente gewünscht wird und noch nicht im AWS Mainframe-Modernisierungsservice verfügbar ist, können Sie eine ausdrückliche Anfrage über Support Product Feature Request (PFR) stellen.

- Damit das Ende des Supports und die Außerbetriebnahme der komponentenspezifischen Versionen von AWS Mainframe Modernization mit dem Ende des Supports des Komponentenlieferanten übereinstimmen.
- Um Kunden etwa ein Jahr vor der Außerbetriebnahme der Hauptversion einer Komponente zu benachrichtigen.

Obwohl wir uns bemühen, diese Richtlinien einzuhalten, können wir in einigen Fällen bestimmte Versionen mit kürzeren Benachrichtigungsfristen früher einstellen. Beispielsweise können wir eine Version mit Sicherheitsproblemen umgehend und mit einem kürzeren Benachrichtigungszeitraum zurückziehen. Wir können auch Nebenversionen vorzeitig zurückziehen, wenn eine Nebenversion erhebliche Fehler oder Sicherheitsprobleme aufweist, die in einer späteren Nebenversion behoben wurden. In dem unwahrscheinlichen Fall, dass solche Fälle eintreten, werden wir die Kunden benachrichtigen und sie über den Plan und den Zeitplan für die Außerbetriebnahme informieren. Bestimmte Umstände können je nach Situation unterschiedliche Zeitpläne vorschreiben.

> **Note**
>
> Wichtige Updates für Komponenten können jederzeit verfügbar gemacht werden. Neue Versionen können beispielsweise aus Sicherheitsgründen oder zur Behebung von Korrekturen für die Produktionsumgebungen umgehend zur Verfügung gestellt werden. Für Anfragen, die über uns gestellt werden Support, legt der Supportplan die Verfahren, den Schweregrad und die Antwortzeiten fest.

Wenn eine Komponentenversion eingestellt wird, verteilt AWS Mainframe Modernization diese Versionen nicht an Kunden für neue Implementierungen. Folglich werden diese Versionen auch nicht von unterstützt. Support Kunden, die bestehende Komponentenbereitstellungen nach dem Datum des Auslaufs ihrer Version ausführen, sollten sich der damit verbundenen Risiken bewusst sein. AWS ist nicht dafür verantwortlich, Sicherheitsupdates, technischen Support oder Hotfixes für veraltete Komponentenversionen bereitzustellen. Außerdem entfernen wir nicht automatisch den Zugriff oder löschen die Ressourcen Ihrer Umgebung. Wir empfehlen Ihnen dringend, alle 3 Monate nach neuen Versionen zu suchen und alle Ihre AWS Mainframe-Modernisierungskomponenten auf aktuelle unterstützte Versionen zu aktualisieren.

# Versions-Upgrade

AWS Mainframe Modernization bietet neuere Versionen aller unterstützten Komponenten, sodass Sie stets über die neuesten Wartungsupdates und -funktionen verfügen. up-to-date Neuere Versionen können Bugfixes, Sicherheitsverbesserungen und andere Verbesserungen für die Komponenten enthalten. Wir empfehlen Ihnen, regelmäßig zu aktualisieren, um von Sicherheitskorrekturen, Bugfixes und Funktionserweiterungen zu profitieren. Wenn AWS Mainframe Modernization eine neue Version veröffentlicht, können Sie wählen, wie und wann Sie Ihre vorhandenen Bereitstellungen aktualisieren möchten. Es gibt zwei Arten von Upgrades: Upgrades von Hauptversionen und Upgrades von Nebenversionen. Generell können in Engine-Hauptversions-Upgrades Änderungen enthalten sein, die nicht mit vorhandenen Anwendungen kompatibel sind. In diesem Fall können für ein Upgrade der Hauptversion erhebliche Änderungen an der Anwendung erforderlich sein. Im Gegensatz dazu beinhaltet ein Upgrade einer kleineren Version Änderungen, die größtenteils abwärtskompatibel mit bestehenden Anwendungen sind. Für ein Upgrade einer kleineren Version sind möglicherweise nur geringe bis gar keine Änderungen erforderlich.

Bevor Sie die Versionsupgrades der Komponenten durchführen, sollten Sie Tests durchführen, die nicht auf Regressionen beruhen. Es hat sich bewährt, DevOps Test- und Bereitstellungspipelines zu verwenden. DevOps Test-Pipelines können im Rahmen von Modernisierungsprojekten erstellt werden und sollten beibehalten werden, um Anwendungstests bei Komponenten-Upgrades und Änderungen des Anwendungscodes zu automatisieren. Sie können bei Upgrades auch blaue/grüne Bereitstellungen oder kanarische Bereitstellungen verwenden. Weitere Informationen zu solchen Bereitstellungen und Change-Management finden Sie unter AWS Well-Architected Reliability Pillar.

# AWS Überblick über die Veröffentlichung von Mainframe Modernization Refactor with Blu Age AWS

Bei AWS Blu Age Runtime folgt die Version einem `Major.Minor.Patch` Muster. Für die AWS Blu Age-Runtime-Version `4.1.0` ist die Hauptversion beispielsweise 4, die Nebenversion ist 1 und die Patch-Version ist 0.

Wir beabsichtigen, neue Hauptversionen von AWS Blu Age Runtime zu veröffentlichen, wenn es wichtige Änderungen an der Laufzeit oder deren Abhängigkeiten gibt. AWS Hauptversionen von Blu Age Runtime werden für mindestens 12 Monate unterstützt, sofern nicht einige häufig auftretende Sicherheitslücken und Sicherheitslücken (CVEs) auftreten. Der Support deckt Fehler in den Runtime-Funktionen ab, wie in unserer Dokumentation erwähnt. Im Fall von Critical und High CVEs in den Abhängigkeiten der Runtime (Spring, Java, Tomcat und andere) wird die Dauer

der Hauptversionsunterstützung auf 6 Monate für High und 3 Monate für Critical CVEs ab dem Veröffentlichungsdatum der neuen Runtime-Version CVEs, die das CVE behebt, reduziert, sofern nicht ausdrücklich anders angegeben.

Wir beabsichtigen, monatlich neue AWS Blu Age-Nebenversionen zu veröffentlichen. Von unseren Kunden wird erwartet, dass sie die Versionen regelmäßig aktualisieren, um die neuesten Sicherheitsupdates, Bugfixes und Funktionserweiterungen zu erhalten. Aktive Projekte, die noch nicht in Produktion sind, müssen die neueste Runtime-Version übernehmen, sobald sie verfügbar ist.

Neue Korrekturen sind in der neuesten Nebenversion für die jeweilige Hauptversion enthalten, bei der ein Problem aufgetreten ist. Wenn Sie neue Fixes benötigen, müssen Sie ein Upgrade auf eine neue Nebenversion durchführen, um diese Fixes anzuwenden.

Patch-Versionen für unterstützte Versionen werden nur zur Behebung kritischer Laufzeitfehler bereitgestellt, die in früheren unterstützten Nebenversionen nicht vorhanden waren.

Alpha-Vorabversionen sind kurzlebige Versionen, die für schnelle Iterationen im Rahmen von Bereitstellungsprojekten zur Verfügung gestellt werden. Korrekturen für Probleme, die in Alpha-Vorabversionen festgestellt wurden, werden in den späteren Nebenversionen bereitgestellt, da für Alpha-Vorabversionen keine Patches bereitgestellt werden.

Veröffentlichungstermine und Details zu den einzelnen Runtime-Versionen finden Sie in der. the section called “AWS Versionshinweise zu Blu Age”

Sicherheitsscans werden von Amazon Inspector durchgeführt.

# Verstehen Sie verwaltete Anwendungen in AWS Mainframe Modernization

Wenn Sie noch nicht mit AWS Mainframe Modernization den folgenden Themen vertraut sind, um loszulegen:

- Was ist AWS Mainframe-Modernisierung?
- Für die AWS Mainframe-Modernisierung gerüstet
- Tutorial: Verwaltete Laufzeit für AWS Blu Age einrichten
- Tutorial: Verwaltete Laufzeit für Rocket Software (ehemals Micro Focus) einrichten

Eine Anwendung 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 Modernization Anwendungen für jeden Workload, den Sie migrieren. Wenn Sie eine AWS Mainframe Modernization Anwendung 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 Rocket Software (ehemals Micro Focus), wenn Sie das Replatforming-Muster verwenden.

Themen

- Erstellen Sie AWS Ressourcen für eine migrierte Anwendung
- Erstellen Sie eine Anwendung AWS Mainframe Modernization
- Stellen Sie eine AWS Mainframe Modernization Anwendung bereit
- Aktualisieren Sie eine Anwendung AWS Mainframe Modernization
- Löschen Sie eine AWS Mainframe Modernization Anwendung
- Reichen Sie Batch-Jobs für AWS Mainframe Modernization Bewerbungen ein
- Stornieren Sie Batch-Jobs für Anwendungen AWS Mainframe Modernization
- Importieren Sie Datensätze für AWS Mainframe Modernization Anwendungen
- Exportieren Sie Datensätze für AWS Mainframe Modernization Anwendungen
- Transaktionen für AWS Mainframe Modernization Anwendungen verwalten

- Konfiguration der von Rocket Software (ehemals Micro Focus) verwalteten Anwendung
- Konfigurieren Sie die verwaltete AWS Blu Age-Anwendung
- AWS Mainframe Modernization Referenz zur Anwendungsdefinition
- AWS Referenz zur Definition von Datensätzen zur Mainframe-Modernisierung

# Erstellen Sie AWS Ressourcen für eine migrierte Anwendung

Um Ihre migrierte Anwendung in ausführen zu können AWS, müssen Sie einige AWS Ressourcen zusammen mit anderen AWS-Services erstellen. 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 Amazon S3 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.

root-folder-name

Der Name ist erforderlich, um die Einschränkungen in der Anwendungsdefinition zu erfüllen, die Sie als Teil der AWS Mainframe Modernization Anwendung erstellen. 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 Mainframe Modernization unterstützt die Verschlüsselung bei der Übertragung in dieser Datenbank. Wenn Sie SSL `sslMode` in Ihrer Datenbank aktivieren, stellen Sie sicher, dass Sie das 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 Replatforming Pattern von Rocket Software 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 `sslMode` in Ihrer Datenbank aktivieren, stellen Sie sicher, dass Sie das

Datenbankgeheimnis zusammen mit den Verbindungsdetails der Datenbank angeben. Sie können einen der folgenden Werte für `sslMode`: `verify-full` `verify-ca`, oder angeben `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" : "*"
   }
```

# Erstellen Sie eine Anwendung AWS Mainframe Modernization

Verwenden Sie die AWS Mainframe Modernization Konsole, um eine AWS Mainframe Modernization Anwendung zu erstellen. Durch das Erstellen einer Anwendung können Sie Aufgaben mit dem migrierten Mainframe-Workload ausführen.

Diese Anleitungen setzen die Durchführung der Schritte unter Für die AWS Mainframe-Modernisierung gerüstet voraus.

## Erstellen einer Anwendung

So erstellen Sie eine Anwendung

1. Öffnen Sie die AWS Mainframe Modernization Konsole 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 Applications die Option Create application aus.
4. Geben Sie auf der Seite Basisinformationen angeben 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 (Rocket) 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**
   >
   > AWS Mainframe Modernization Verschlüsselt Ihre Daten standardmäßig mit einem AWS KMS Schlüssel, der Ihnen AWS Mainframe Modernization gehört und der für Sie verwaltet wird. Sie können sich jedoch dafür entscheiden, einen vom Kunden verwalteten AWS KMS Schlüssel zu verwenden.

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 Anwendungsdefinition von Rocket Software (ehemals Micro Focus).
12. Wählen Sie Weiter aus.
13. Überprüfen Sie auf der Seite Überprüfen und erstellen die von Ihnen eingegebenen Informationen und wählen Sie dann Anwendung erstellen.

# Stellen Sie eine AWS Mainframe Modernization Anwendung bereit

Verwenden Sie die AWS Mainframe Modernization Konsole, um eine AWS Mainframe Modernization Anwendung bereitzustellen. Sie müssen Ihre Anwendungen in einer Laufzeitumgebung bereitstellen, bevor Sie Aufgaben ausführen können.

Diese Anleitungen setzen die Durchführung der Schritte unter Für die AWS Mainframe-Modernisierung gerüstet voraus.

## Stellen Sie eine Anwendung bereit

Um eine AWS Mainframe Modernization Anwendung 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 Konsole 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.
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 Konsole 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 Anwendung AWS Mainframe Modernization

Verwenden Sie die AWS Mainframe Modernization Konsole, um eine AWS Mainframe Modernization Anwendung zu aktualisieren. Durch das Aktualisieren einer Anwendung wird eine neue Version der Anwendung erstellt.

Diese Anleitungen setzen die Durchführung der Schritte unter Für die AWS Mainframe-Modernisierung gerüstet voraus.

## Aktualisieren einer Anwendung

Eine AWS Mainframe Modernization Anwendung kann mehrere Versionen haben, jede mit ihrer eigenen Anwendungsdefinition. 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 Konsole 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 Anwendungsdefinition von Rocket Software (ehemals Micro Focus).

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 Modernization Anwendung bereit.

# Löschen Sie eine AWS Mainframe Modernization Anwendung

Sie können eine AWS Mainframe Modernization Anwendung mithilfe der AWS Mainframe Modernization Konsole aus einer Umgebung löschen.

Diese Anleitungen setzen die Durchführung der Schritte unter Für die AWS Mainframe-Modernisierung gerüstet voraus.

## Löschen einer Anwendung

Wenn Sie eine AWS Mainframe Modernization Anwendung löschen müssen und sie 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 Konsole 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.

# Reichen Sie Batch-Jobs für AWS Mainframe Modernization Bewerbungen ein

Darin können AWS Mainframe Modernization 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 senden, AWS Mainframe Modernization wird eine separate Batch-Job-Ausführung erstellt. 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 Modernization Konsole, um die Datensätze zu importieren. Weitere Informationen finden Sie unter Importieren Sie Datensätze für AWS Mainframe Modernization Anwendungen.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter Für die AWS Mainframe-Modernisierung gerüstet und in abgeschlossen habenErstellen Sie eine Anwendung AWS Mainframe Modernization.

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 Konsole 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.

   > **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 von Batch-Jobs ist in den folgenden Engine-Versionen verfügbar:
>
> - Micro Focus (Rocket) Environment Engine-Versionen 8.0.6 oder höher. Sie müssen außerdem ein EFS- oder FSx Dateisystem an Ihre Umgebung angehängt haben.
> - AWS Blu Age Environment Engine-Versionen 4.3.0 oder höher. Sie müssen auch ein EFS- oder FSx Dateisystem angehängt haben, wenn es sich um eine HA-Umgebung handelt.

1. Öffnen Sie die AWS Mainframe Modernization Konsole unter 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. Für die Micro Focus (Rocket) Environment Engine und die AWS Blu Age Environment Engine können Sie Folgendes tun:

   - Für die Micro Focus (Rocket) Environment Engine können Sie entweder von Anfang an neu starten oder mit steps oder procsteps neu starten wählen.
     - Mit der Option „Von Anfang neu starten“ können Sie alle Schritte eines Batch-Jobs von Anfang an neu starten.
     - Mit der Option Mit Schritten oder procsteps neu starten können Sie einen bestimmten Schritt oder Verfahrensschritt auswählen, den Sie neu starten möchten, und optional einen Schritt oder Verfahrensschritt, den Sie beenden möchten.

     > **Note**
     >
     > Der Endschritt oder Procstep muss größer oder gleich der Startschritt- oder Procstep-Nummer sein.

   - Bei der AWS Blu Age Environment Engine können Sie entweder die letzte Ausführung eines Batch-Jobs von einem zuvor fehlgeschlagenen JCL/PROC-Schritt aus neu starten oder einen verzögerten Neustart durchführen, indem Sie zuvor erfolgreiche Schritte umgehen.
     - Sie können einen bestimmten Schrittnamen wählen, den Sie neu starten möchten.
     - Optional können Sie „Schritt überspringen“ verwenden, um den ausgewählten Schritt zu umgehen und mit dem nächsten Schritt im Workflow neu zu starten.
7. Wählen Sie Auftrag absenden.

# Stornieren Sie Batch-Jobs für Anwendungen AWS Mainframe Modernization

Darin können AWS Mainframe Modernization Sie Batch-Jobs für Ihre Anwendungen stornieren. Sie können Details zur Ausführung von Batch-Jobs überprüfen. Jedes Mal, wenn Sie einen Batch-Job einreichen, AWS Mainframe Modernization wird eine separate Batch-Job-Ausführung erstellt. 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.

## Stornieren Sie einen Batch-Job

Wenn Sie einen Batch-Job stornieren, wird damit nicht ein Batch-Job gelöscht, sondern die Ausführung von Aufgaben für diesen Batch-Job. Sie können weiterhin die Details Ihres Batch-Jobs anzeigen.

Um einen Batch-Job abzubrechen

1. Öffnen Sie die AWS Mainframe Modernization Konsole unter https://console.aws.amazon.com/m2/.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der sich die Anwendung für Ihre Batch-Jobs befindet.
3. Suchen Sie in der Batch-Job-Liste den Batch-Job, den Sie stornieren möchten, und wählen Sie ihn aus.
4. Wählen Sie „Aktionen“ und anschließend „Auftrag stornieren“.
5. Wählen Sie Batch-Job stornieren.

Dadurch werden alle Batch-Job-Aufgaben abgebrochen, die Sie für die Ausführung geplant hatten.

# Importieren Sie Datensätze für AWS Mainframe Modernization Anwendungen

Mit können AWS Mainframe Modernization Sie Datensätze importieren, um sie mit Ihren Anwendungen zu verwenden. Sie können die zu importierenden Datensätze in einer JSON-Datei angeben, die in einem Amazon S3 S3-Bucket gespeichert ist, oder Sie können die Werte für die Datensatz-Konfiguration 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 Modernization Konsole, um Datensätze für eine AWS Mainframe Modernization Anwendung zu importieren.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter Für die AWS Mainframe-Modernisierung gerüstet und in abgeschlossen habenErstellen Sie eine Anwendung AWS Mainframe Modernization.

## Importieren Sie einen Datensatz

Um einen Datensatz zu importieren

1. Öffnen Sie die AWS Mainframe Modernization Konsole 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. Spezifische Definitionsdetails finden Sie unterthe 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.

# Exportieren Sie Datensätze für AWS Mainframe Modernization Anwendungen

Mit können AWS Mainframe Modernization Sie Datensätze exportieren, um sie mit Ihren Anwendungen zu verwenden. Sie können die zu exportierenden 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 exportiert haben, können Sie die Details der Exportaufgabe überprüfen, um sicherzustellen, dass die gewünschten Datensätze exportiert wurden.

Verwenden Sie die AWS Mainframe Modernization Konsole, um Datensätze für eine AWS Mainframe Modernization Anwendung zu exportieren.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die Schritte unter Für die AWS Mainframe-Modernisierung gerüstet und in abgeschlossen habenErstellen Sie eine Anwendung AWS Mainframe Modernization.

## Exportieren Sie einen Datensatz

Um einen Datensatz zu exportieren

1. Öffnen Sie die AWS Mainframe Modernization Konsole 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 exportieren möchten.
4. Wählen Sie auf der Seite mit den Anwendungsdetails die Option Datensätze aus.
5. Wählen Sie Export 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. Weitere Informationen finden Sie unter the section called “Referenz zur Definition von Datensätzen”.

     Geben Sie den Datensatznamen, den externen Amazon S3 S3-Speicherort und 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.

# Transaktionen für AWS Mainframe Modernization Anwendungen verwalten

Mit können AWS Mainframe Modernization 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 Für die AWS Mainframe-Modernisierung gerüstet und in abgeschlossen habenErstellen Sie eine Anwendung AWS Mainframe Modernization.

## Transaktionen für Anwendungen verwalten

Um Transaktionen für Anwendungen zu verwalten

1. Öffnen Sie die AWS Mainframe Modernization Konsole 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 sortiert anzeigen.

   - Mit Transaktionsressourcen können Sie den Ressourcentyp anhand von Dateidefinitionen, Transaktionsdefinitionen, Programmdefinitionen oder Definitionen für transiente Datenwarteschlangen auswählen.

     > **Note**
     >
     > Der AWS Mainframe Modernization Dienst 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 Replatform Engine von Rocket Software können Sie bis zu vier Listen in einer SIT angeben.
   - SIT (System Initialization Table) zeigt alle verfügbaren Transaktionskonfigurationen an. Sie können SITs anhand der Eigenschaften (Name, Beschreibung und Startlisten) suchen. Sie können auch Listen auswählen, die Sie mit Ihrer ausgewählten SIT verknüpfen möchten.

     > **Note**
     >
     > SITs gelten nur für die Rocket Software Replatform Engine.

5. Wählen Sie eine Transaktionsressource aus, um alle Ressourceninformationen anzuzeigen. Sie können auch alle Attribute anzeigen, die mit Ihrer Transaktionsressource verknüpft sind.

# Konfiguration der von Rocket Software (ehemals Micro Focus) verwalteten Anwendung

Sie können Ihre Anwendungen mit der Runtime Engine von Rocket Software konfigurieren, um zusätzliche Eigenschaften, einschließlich Integrationen, anzupassen.

Themen


- Unterstützte Integrationen von Drittanbietern für Rocket Software
  - Drucker


## Unterstützte Integrationen von Drittanbietern für Rocket Software

Um Integrationen von Drittanbietern nutzen zu können, muss Ihre von AWS Mainframe Modernization verwaltete Umgebung eine Engine-Version von Rocket Software verwenden, die diese Art von Konfiguration unterstützt. Engine-Versionen mit dem Postfix R (z. B. Version 9.0.9.R) werden unterstützt. Das bedeutet, dass die Engine-Version 9.0.9.R die Client-Installation für die Party-Integrationen unterstützt, 9.0.9 jedoch nicht.

### Drucker

Druckerressourcen werden über die Rocket Software-Anwendungsdefinition konfiguriert, wie im Abschnitt beschrieben. the section called “Drucker — optional”

Eine Druckerdefinition kann ein benutzerdefiniertes oder vom Service bereitgestelltes Ausgangsmodul für den Drucker definieren. Einige Beispiele für mögliche Konfigurationen von Ausgangsmodulen sind:

1. Beispiel für das Laden des Dienstes, der binär bereitgestellt wird.

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

```
    }
},
...
```

2. Beispiel für die Bereitstellung von Binärdateien aus S3.

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

3. Beispiel für die Bereitstellung von Binärdateien aus EFS.

> **Note**
>
> Um EFS-Mount verwenden zu können, muss es bei der Erstellung der Umgebung zusammen mit einigen zusätzlichen Werten, die festgelegt werden müssen, angehängt `program-path` werden, z.

```
...
"batch-settings": {
    "jes-printers": [
        {
            "name": "p3",
            "classes": [
                "EF"
            ],
            "description": "Using binary from customer provided exit module on EFS
 Mount",
            "exit-module": {
                "name": "efsExit"
            }
        }
    ],
    "program-path": "$EFS_MOUNT/path/to/directory/containing/binaries/"
},
"runtime-settings": {
    "environment-variables": {
        "EFS_MOUNT": "/m2/mount/efs"
    }
```

```
}
...
```

LRS-Warteschlange — optional

Um die LRS-Warteschlange verwenden zu können, müssen Sie eine Rocket Software-Engine verwenden, die Artefakte von Drittanbietern unterstützt (d. h. Engines mit `.R` der Endung). Zusätzlich zur Einrichtung eines Druckers mit einem Ausgangsmodul, auf das `lrsprte6` als Eintragsname des Ausgangsmoduls verwiesen wird, benötigt die LRS-Warteschlange eine zusätzliche Umgebungsvariable, wie sie im bereits vorhandenen Block „Runtime-Einstellungen“ der Rocket Software-Anwendungsdefinition definiert ist.

LRSQ_ADDRESS

(Erforderlich) Gibt die LRS-Serveradresse an, an die das LRSQ-Print-Exit-Modul gesendet werden soll.

LRS-Drucker — Für die Konfiguration eines LRS-Druckers ist die Definition eines JES-Druckers erforderlich, wie im Abschnitt beschrieben. the section called “Drucker — optional”

Darüber hinaus muss LRSQ_ADDRESS als Teil des `runtime-settings` Felds in der Anwendungsdefinition angegeben werden.

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

# Konfigurieren Sie die verwaltete AWS Blu Age-Anwendung

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, lesen Sie den the section called “Struktur der von AWS Blu Age verwalteten Anwendungen” Abschnitt über die Gesamtstruktur einer modernisierten AWS Blu Age-Anwendung.

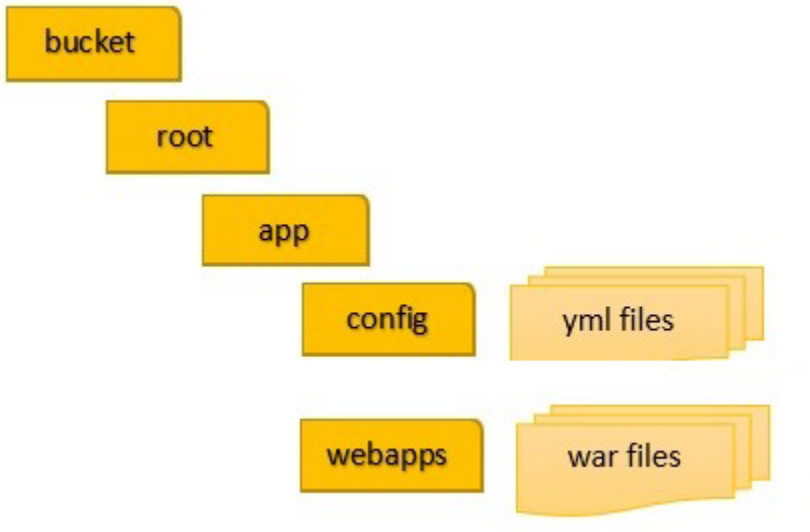
Themen


- Struktur der von AWS Blu Age verwalteten Anwendungen

- Zugriff auf Dienstprogramme für verwaltete Anwendungen konfigurieren
- Fügen Sie Konfigurationseigenschaften für die verwaltete Anwendung mit AWS Blu Age Engine hinzu

## Struktur der von AWS Blu Age verwalteten Anwendungen

Wenn Sie das AWS Blu Age-Refactoring-Muster 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.

lnk

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 der Java-Optionen einer Anwendung

Um einige Java-Optionen für die Anwendung zu verwalten, fügen Sie dem `application-name` Ordner eine Eigenschaftendatei mit dem Namen `tomcat.properties` hinzu. Diese Datei kann drei Eigenschaften haben:`xms`, die den minimalen Java-Speicherverbrauch angibt`xmx`, die den maximalen Java-Speicherverbrauch angibt und`dnscachettl`, die Cachedauer für DNS-Auflösungen verwaltet. Das Folgende ist ein Beispiel für den Inhalt einer gültigen `tomcat.properties` Datei.

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

Die Werte, die Sie für die ersten 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.

Der Wert für die dritte Eigenschaft stellt die Cachedauer in Sekunden dar und kann den Wert -1 (für immer zwischenspeichern) oder einen Wert zwischen 0 (nie zwischenspeichern) und 999 haben. Im Kontext verwalteter Anwendungsbereitstellungen ist der Standardwert -1.

# Zugriff auf Dienstprogramme für verwaltete Anwendungen konfigurieren

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
```

sysPunchEncoding

(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 Datenquelle

Einige datenbankbezogene Dienstprogramme, wie LOAD und UNLOAD, erfordern 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:

Primäre Datenquelle

Dies ist die primäre Datenbank für Geschäftsanwendungen.

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
```

spring.aws.client.datasources.primary.type

(Optional) Gibt den vollqualifizierten Namen der zu verwendenden Verbindungspool-Implementierung an. Der Standardwert ist `com.zaxxer.hikari.HikariDataSource`.

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

Wenn der Typ der primären Datenquelle ist`com.zaxxer.hikari.HikariDataSource`, können Sie zusätzliche Eigenschaften wie folgt angeben:

spring.datasource.primary. [Eigenschaftsname]

(Optional) Sie können dieses Format verwenden, um zusätzliche Eigenschaften für die Konfiguration einer primären Datenquellenverbindungspool-Implementierung anzugeben.

Im Folgenden finden Sie ein Beispiel für eine primäre Datenquelle vom Typ`com.zaxxer.hikari.HikariDataSource`.

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

Andere Datenquellen für Versorgungsunternehmen

Zusätzlich zur primären Datenquelle können Sie weitere Datenquellen für Versorgungsunternehmen bereitstellen.

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

(Optional) Gibt die Liste der Namen der Dienstprogramm-Datenquellen an.

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

spring.aws.client.utility.pgm.datasources. [DSname] .geheim

(Optional) Gibt den geheimen ARN in SSM an, der die Datenquelleneigenschaften hostet. Geben Sie [dsname] in der Liste der unter angegebenen Namen ein. `spring.aws.client.utility.pgm.datasources.names`

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

spring.aws.client.utility.pgm.datasources. [DSName] .DB-Name

(Optional) Gibt den Namen der Zieldatenbank an, wenn der Datenbankname nicht mithilfe der Eigenschaft direkt im geheimen Datenbankschlüssel angegeben ist. `dbname` Geben Sie [dsname] in der Liste der in `spring.aws.client.utility.pgm.datasources.names` angegebenen Namen ein.

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

spring.aws.client.utility.pgm.datasources. [DSname] .typ

(Optional) Gibt den vollqualifizierten Namen der zu verwendenden Verbindungspool-Implementierung an. Der Standardwert ist `com.zaxxer.hikari.HikariDataSource`. Geben Sie [dsname] in der Liste der unter angegebenen Namen ein. `spring.aws.client.utility.pgm.datasources.names`

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

Wenn der Utility-Datenquellentyp ist`com.zaxxer.hikari.HikariDataSource`, können Sie zusätzliche Eigenschaften wie folgt angeben:

spring.datasource. [DNS-Name]. [Eigenschaftsname]

(Optional) Gibt eine Sammlung zusätzlicher Eigenschaften zur Konfiguration einer Implementierung eines Verbindungspools für Dienstprogrammdatenquellen an. Geben Sie [dsname] in der Liste der unter angegebenen Namen ein. `spring.aws.client.utility.pgm.datasources.names` Geben Sie die Eigenschaften im folgenden Format an: `property_name : value`

Im Folgenden finden Sie ein Beispiel für zusätzliche Versorgungsdatenquellen des Typs`com.zaxxer.hikari.HikariDataSource`:

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

treatLargeNumberAsInteger

(Optional) Bezieht sich auf die Besonderheiten der Oracle-Datenbank-Engine und die Verwendung von DSNTEP2/DSNTEP4 Utilities. 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 von 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 „`0`bis9“ 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 (AS400) -Plattformen stammen.

```
zonedMode:EBCDIC_STRICT
```

jcl.type

(Optional) Gibt den älteren Typ modernisierter JCL-Skripts an. Das IDCAMS-Hilfsprogramm verwendet diese Einstellung, um den Rückgabecode anzupassen, wenn es sich bei der aufrufenden JCL um einen Typ handelt. `vse` 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
```

sqlCodePointUmschalten

(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 Nullindikatorbyte 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"
```

- whenNotNull: 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"
```

useDatabaseConfiguration

(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 gesetzt`true`, werden die `pgmTimestampFormat` Eigenschaften `pgmDateFormatpgmTimeFormat`, und aus der Hauptkonfigurationsdatei () `application-main.yml` verwendet.
- Wenn auf gesetzt`false`, 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
```

varCharIsNull

(Optional) Gibt an, wie eine Varchar-Spalte mit leerem Inhalt behandelt werden soll, die keine NULL-Werte zulassen. Der Standardwert ist `false`.

Wenn Sie diesen Wert auf setzen`true`, 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
```

sqlCodePointUmschalten

(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 verwaltete Anwendung mit AWS Blu Age Engine hinzu

Sie können dem `config` Ordner für Ihre umgestaltete Anwendung eine Datei hinzufügen, mit der Sie auf neue Funktionen der AWS Blu Age Runtime Engine zugreifen können. 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
- Gapwalk-Batchscript-Eigenschaften
- 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`

Type: Zeichenkette

Standard: CP1145

Größe der Eingabedatei überprüfen

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 gesetzt`true`, 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

Datenvereinfacher. onInvalidNumericDaten

Optional. Wie man reagiert, wenn man ungültige numerische Daten dekodiert. Zulässige Werte sind`reject`,`toleratespaces`,`toleratespaceslowvalues`,`toleratemost`.

Type: Zeichenkette

Standard: Ablehnen

defaultKeepExistingDateien

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.

Type: Zeichenkette

Standard: null

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

blockSizeDefault

Optional. Die Standardblockgröße, die für BDW-Bytes verwendet werden soll.

Typ: Zahl

Standard: 32760

```
blockSizeDefault: 32760
```

Für HR

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.

Type: Zeichenkette

Standard: &, @,%%

abfrage. useConcatCondition

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

sctThreadLimit

Optional. Das Thread-Limit für das Auslösen von Skripten.

Typ: Zahl

Standard: 5

sqlCodePointUmschalten

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

sqlIntegerOverflowZulässig

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

stepFailWhenAbend

Optional. Gibt an, ob ein Abend ausgelöst werden soll, wenn ein Schritt fehlschlägt oder die Ausführung abschließt.

Typ: Boolesch

Standard: true

stopExecutionWhenProgNotFound

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

uppercaseUserInput

Optional. Gibt an, ob Benutzereingaben in Großbuchstaben erfolgen müssen.

Typ: Boolesch

Standard: true

Benutze die Steuerung MVariable

Optional. Gibt an, ob die Control-M-Spezifikation für den Variablenersatz verwendet werden soll.

Typ: Boolesch

Standard: false

jcl.CheckPoint.ExpireTimeout

Optional. Gibt die Zeitdauer für die Beibehaltung von JCL-Checkpoints im Persistenzanbieter oder in der In-Memory-Registrierung an.

Typ: Zahl

Standard: -1

jcl.checkpoint. expireTimeoutUnit

Optional. Gibt die Zeitdauereinheit für die `jcl.checkpoint.expireTimeout` Eigenschaft an. Unterstützte Enum-Konstantenwerte:`java.util.concurrent.TimeUnit`.

Type: Zeichenkette

Standard: SECONDS

Gapwalk-Batchscript-Eigenschaften

encoding

Optional. Die in Batchscript-Projekten verwendete Kodierung (nicht bei Groovy). Erwartet eine gültige Kodierung`CP1047`,,,... `IBM930` `ASCII` `UTF-8`

Type: Zeichenkette

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;`.

Type: Zeichenkette

Standard: null

```
managers.trancode: OR12:MYDIALOG1
```

Eigenschaften des Gapwalk CL-Befehls

Befehle aus

Optional. Liste der auszuschaltenden Befehle, getrennt durch Komma. Zulässige Werte sind `PGM_BASICRCVMSG,SNDRCVF,CHGVAR,QCLRDTAQ,RTVJOBA,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.

Type: Zeichenkette

Standard: null

spring.datasource.primary.jndi-name

Optional. Die primäre JNDI-Datenquelle (Java Naming And Directory Interface).

Type: Zeichenkette

Standard: jdbc/primary

Zonierter Modus

Optional. Der Modus für die Kodierung oder Dekodierung von Datentypen in Zonen. Zulässige Werte sind`EBCDIC_STRICT`//`EBCDIC_MODIFIED`. `AS400`

Type: Zeichenkette

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...`

Type: Zeichenkette

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`.

Type: Zeichenkette

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 (). `,` Beispiel: `ims.programs: PCP008,PCT008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;`

Type: Zeichenkette

Standard: null

jhdb.CheckPointPath

Optional. `jhdb.checkpointPersistence`lst 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.

Type: Zeichenkette

Standard: Datei:. /setup/

jhdb.CheckPointPersistenz

Optional. Der Checkpoint-Persistenzmodus. Zulässige Werte sindnone//add. end Wird verwendetadd, um Checkpoints beizubehalten, wenn ein neuer erstellt und der Registrierung hinzugefügt wird. Wird verwendetend, 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.

Type: Zeichenkette

Standard: keiner

jhdb.configuration.context.encoding

Optional. Die JHDB-Kodierung (Java Hierarchical Database). Erwartet eine gültige KodierungszeichenfolgeCP1047,,,... `IBM930` `ASCII` `UTF-8`

Type: Zeichenkette

Standard: CP297

jhdb. identificationCardData

Optional. Wird verwendet, um einige „Bediener-Identifikationskartendaten“ fest in das durch den CARD-Parameter angegebene MID-Feld zu codieren.

Type: Zeichenkette

Standard: ""

jhdb.lterm

Optional. Ermöglicht es Ihnen, bei einer IMS-Emulation eine gemeinsame logische Terminal-ID zu erzwingen. Wenn nicht gesetzt, wird sessionId verwendet.

Type: Zeichenkette

Standard: null

jhdb.metadata.extrapath

Ein Konfigurationsparameter, der einen zusätzlichen, laufzeitspezifischen Stammordner für psbs- und dbds-Ordner angibt.

Type: Zeichenkette

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. limitJoinUsage

Optional. Gibt an, ob der Parameter Limit Join-Verwendung in RDBMS-Diagrammen verwendet werden soll.

Typ: Boolesch

Standard: true

jhdb. use-db-prefix

Optional. Gibt an, ob ein Datenbankpräfix in der hierarchischen Navigation für ein RDBMS aktiviert werden soll.

Typ: Boolesch

Standard: true

Gapwalk JICS-Eigenschaften

jics.data. dataJsonInitStandort

Optional. Speicherort der JSON-Datei, die vom Analyzer anhand der CSD-Analyse erstellt und zur Initialisierung der JICS-Datenbank verwendet wurde,

Type: Zeichenkette

Standard: ""

jics.db. dataScriptLocation

Optional. Speicherort des Skripts initJics.sql, das von Analyzer anhand der Analyse von CSD-Exporten vom Mainframe erstellt wurde.

Type: Zeichenkette

Standard: ""

jics.db. dataTestQueryStandort

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.

Type: Zeichenkette

Standard: ""

jics.db. ddlScriptLocation

Optional. Der Speicherort des Mics-DDL-Skripts. Ermöglicht es Ihnen, das Jics-Datenbankschema mithilfe eines SQL-Skripts zu initiieren.

Type: Zeichenkette

Standard: ""

```
jics.db.ddlScriptLocation: ./jics/sql/jics.sql
```

jics.db. schemaTestQueryStandort

Optional. Speicherort der SQL-Datei, die eine eindeutige Abfrage enthalten soll, die die Anzahl der Objekte im Jics-Schema zurückgibt (falls vorhanden).

Type: Zeichenkette

Standard: ""

Jics. runUnitLauncherPool.aktivieren

Optional. Gibt an, ob der Run Unit Launcher-Pool in JICS aktiviert werden soll.

Typ: Boolesch

Standard: false

JICS. runUnitLauncherPool.Größe

Optional. Die Poolgröße des Run-Unit-Launchers in JICS.

Typ: Zahl

Standard: 20

Jics. runUnitLauncherPool.Validierungsintervall

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.

Type: Zeichenkette

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.

Type: Zeichenkette

Standard: eu-west-3

TaskExecutor. allowCoreThreadTimeOut

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. corePoolSize

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. maxPoolSize

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

dateTimeFormat

Optional. dateTimeFormat In wird beschrieben, wie der Datums- und Uhrzeitstempeltyp einer Datenbank in Data Simplifier-Entitäten übertragen wird. Zulässige Werte sind///ISOEURUSALOCAL

Type: Zeichenkette

Standard: ISO

dbDateFormat

Optional. Das Zieldatumsformat der Datenbank.

Type: Zeichenkette

Standard: yyyy-MM-dd

dbTimeFormat

Optional. Das Zielzeitformat der Datenbank.

Type: Zeichenkette

Standard: hh:mm:ss

dbTimestampFormat

Optional. Das Zeitstempelformat für das Datenbankziel.

Type: Zeichenkette

Standard: hh:mm:ss.ssssss yyyy-MM-dd

Größe abrufen

Optional. Der FetchSize-Wert für Cursor. Wird verwendet, wenn Daten mithilfe von Chunks mithilfe von Load/Unload-Utils abgerufen werden.

Typ: Zahl

Standard: 10

Deaktivieren erzwingen SQLTrim StringType

Optional. Gibt an, ob das Kürzen aller SQL-String-Parameter deaktiviert werden soll.

Typ: Boolesch

Standard: false

localDateFormat

Optional. Liste der lokalen Datumsformate. Trennen Sie jedes Format durch|.

Type: Zeichenkette

localTimeFormat

Optional. Liste der lokalen Zeitformate. Trennen Sie jedes Format durch|.

Type: Zeichenkette

localTimestampFormat

Optional. Liste der lokalen Zeitstempelformate. Trennen Sie jedes Format durch|.

Type: Zeichenkette

Standard:

pgmDateFormat

Optional. Das in den Programmen verwendete Datums- und Uhrzeitformat.

Type: Zeichenkette

Standard: yyyy-MM-dd

pgmTimeFormat

Optional. Das für die Ausführung von PGM-Programmen (Programmen) verwendete Zeitformat.

Type: Zeichenkette

Standard: HH.MM.ss

pgmTimestampFormat

Optional. Das Zeitstempelformat.

Type: Zeichenkette

Standard: .mm.ss.SSSSSS yyyy-MM-dd-HH

Eigenschaften des Gapwalk-Hilfsprogramms

jcl.type

Fakultativ. `.jcl`Dateityp. 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.

Type: Zeichenkette

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`.

Type: Zeichenkette

Standard: rdw

load.BatchSize

Optional. Die Batchgröße des Ladedienstprogramms.

Typ: Zahl

Standard: 0

load.format.dbDate

Optional. Das zu verwendende Datenbankformat des Load Utilitys.

Type: Zeichenkette

Standard: yyyy-MM-dd

load.format.dbTime

Optional. Die zu verwendende Uhrzeit der Load Utility-Datenbank.

Type: Zeichenkette

Standard: hh:mm:ss

load.format.LocalDate

Optional. Das zu verwendende lokale Datumsformat des Ladedienstprogramms.

Type: Zeichenkette

Standard: dd/MM/yyyy dd.mm.yyyy| |yyyy-MM-DD

load.format.LocalTime

Optional. Das zu verwendende lokale Zeitformat des Ladedienstprogramms.

Type: Zeichenkette

Standard: hh:mm:ss|hh.mm.ss

laden. sqlCodePointSchicht

Optional. Das SQL-Code-Programm Pointshift for Load. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank Postgresql DB2 ist.

Typ: Zahl

Standard: 0

sysPunchEncoding

Optional. Der Zeichensatz für die Syspunch-Kodierung. Unterstützte Werte sind/`Cp1047`. `ASCII`

Type: Zeichenkette

Standard: ASCII

treatLargeNumberAsInteger

Optional. Gibt an, ob große Zahlen als behandelt werden sollen. `Integer` 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.

Type: Zeichenkette

Standard: Leerzeichen

unload.fetchSize

Optional. Ermöglicht die Einstellung der Abrufgröße bei der Bearbeitung von Cursorn im Hilfsprogramm zum Entladen.

Typ: Zahl

Standard: 0

unload.format.date

Optional. Wenn aktiviert, `unload.useDatabaseConfiguration` das Datumsformat, das im Entladeprogramm verwendet werden soll.

Type: Zeichenkette

Standard: MM/dd/yyyy

unload.format.time

Optional. Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitformat, das im Entladeprogramm verwendet werden soll.

Type: Zeichenkette

Standard: HH.MM.ss

unload.format.timestamp

Optional. Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitstempelformat, das im Entladeprogramm verwendet werden soll.

Type: Zeichenkette

Standard: .mm.ss.SSSSSS yyyy-MM-dd-HH

.nbi entladen. whenNotNull

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 entladen. writeNullIndicator

Optional. Gibt an, ob der Null-Indikator in der Entladeausgabedatei ausgegeben werden soll.

Typ: Boolesch

Standard: false

entladen. sqlCodePointSchicht

Optional. Das SQL-Code-Hilfsprogramm Pointshift for Unload. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank Postgresql DB2 ist.

Typ: Zahl

Standard: 0

entladen. useDatabaseConfiguration

Optional. Gibt an, ob die Datums- oder Uhrzeitkonfiguration aus application-main.yml im Unload-Hilfsprogramm verwendet werden soll.

Typ: Boolesch

Standard: false

entladen. varCharIsNull

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. stand-by-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

warmUpCache

Optional. Gibt an, ob beim Serverstart alle Daten der Datenkommunikationstabelle in einen Aufwärmcache geladen werden sollen.

Typ: Boolesch

Standard: false

# AWS Mainframe Modernization Referenz zur Anwendungsdefinition

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 die Anwendungsdefinitionen von AWS Blu Age und Rocket Software (ehemals Micro Focus) beschrieben und alle erforderlichen und optionalen Elemente beschrieben.

Inhalt

- Allgemeiner Kopfbereich

- Überblick über den Definitionsbereich
- AWS Beispiel für eine Blu-Age-Anwendungsdefinition
- AWS Einzelheiten zur Blu Age-Definition
  - Zuhörer — erforderlich
  - AWS Blu Age-Anwendung — erforderlich
  - BluSAM — fakultativ
  - AWS Blu Age-Nachrichtenwarteschlangen — optional
  - AWS EFS-Konfiguration für Blu-Age-Anwendungsspeicher — optional
- Anwendungsdefinition von Rocket Software (ehemals Micro Focus)
- Einzelheiten zur Definition von Rocket Software
  - Listener — erforderlich
  - Speicherorte von Datensätzen — erforderlich
  - Amazon Cognito Cognito-Authentifizierungs- und Autorisierungshandler — optional
  - LDAP- und Active Directory-Handler — optional
  - Batch-Einstellungen — erforderlich
  - CICS-Einstellungen — erforderlich
  - Drucker — optional
  - XA-Ressourcen — optional
  - Laufzeiteinstellungen — optional


## 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 je nach Bedarf in der JSON-Anwendungsdefinition auf den Quellspeicherort zu verweisen.

Quelltyp

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, AWS Mainframe Modernization erkennt es 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. Die folgenden Abschnitte enthalten Beispiele für modulspezifische Anwendungsdefinitionen für beide Engines.

## 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 verwenden, mit Ausnahme der bekannten Ports von 0 bis 1023. Wir empfehlen, den Bereich von 8192 bis 8199 zu verwenden. Stellen Sie sicher, dass an diesem Port keine anderen Listener oder Anwendungen ausgeführt werden.

Typ

(Erforderlich) Derzeit `http` wird nur 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": "${s3-source}/murachs-v6/",
            "files-directory": "/m2/mount/myfolder",
            "enable-jics": <true|false>,
            "enable-batch-restart": <true|false>,
            "shared-app-location": "${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 FSx Amazon-Mount-Point-Einrichtung auf Umgebungsebene sein. Der Unterordner muss einem geeigneten Benutzer gehören, damit er von der darin ausgeführten Blu Age-Anwendung verwendet werden kann. AWS Mainframe Modernization Um dies zu erreichen, `3001` muss beim Anhängen des Laufwerks an eine EC2 Linux-Amazon-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.

enable-batch-restart

(Optional) Gibt an, ob die Neustartfunktion für Batch-Jobs aktiviert werden soll. Standardwert "false". Weitere Informationen zu Konfigurationen für den Batch-Neustart finden Sie

unter Eigenschaften der AWS Blu Age-Engine mit dem Präfix `jcl.checkpoint` unter Konfigurationseigenschaften für die verwaltete Anwendung mit der AWS Blu Age-Engine.

shared-app-location

(Optional) Weiterer Speicherort 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`— (Optional) 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`— (Optional) 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`— (Optional) 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",
      "ssl-cipher": "*TLS12ORHIGHER",
```

```
      "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  },
  {
      "product-type": "JMS-MQ",
      "queue-manager": "QMgr2",
      "channel": "mqChannel2",
      "hostname": "mqserver-host2",
      "port": 1412,
      "user-id": "app-user2",
      "ssl-cipher": "*TLS12ORHIGHER",
      "secret-manager-arn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:sample/mq/test-279PTa"
  }
]
```

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-Portnummer an, auf der der Server lauscht.

Benutzer-ID

(Optional) Gibt die Benutzerkonto-ID an, die Nachrichtenwarteschlangenoperationen auf dem angegebenen Kanal ausführen darf.

SSL-Cipher

(Optional) Gibt die SSL-Verschlüsselungsspezifikation für die Verbindung an.

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"
           }
]
```

file-system-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 Bereitstellungspunkt für das Dateisystem auf Anwendungsebene. Dieser muss sich vom Speichermountpunkt auf Umgebungsebene unterscheiden.

access-point-id

(Erforderlich) Die von Amazon EFS zugewiesene ID des Access Points. 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.

## Anwendungsdefinition von Rocket Software (ehemals Micro Focus)

Der folgende Abschnitt mit einer Beispieldefinition bezieht sich auf die Runtime-Engine von Rocket Software 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": "ADRDSSU"}
            ]
        },
        "cics-settings": {
            "binary-file-location": "${s3-source}/cics/binaries",
            "csd-file-location": "${s3-source}/cics/def",
            "system-initialization-table": "BNKCICV"
        },
        "jes-printers": [
           {
             "name": "printerName",
              "classes": [
                 "A",
                 "B"
             ],
             "description": "printer desc....",
             "exit-module": {
                 "name": "lrsprte6"
                }
             }
         ],
        "xa-resources" : [{
            "name": "XASQL",
            "secret-manager-arn": "arn:aws:secrets:1234:us-east-1:secret:123456",
            "xa-connection-type": "postgres",
            "module": "${s3-source}/xa/ESPGSQLXA64.so"
```

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

## Einzelheiten zur Definition von Rocket Software

Der Inhalt des Definitionsabschnitts der Rocket Software-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. Zuhörer 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. AWS Mainframe Modernization Unterstützt derzeit nur Datensätze aus einer einzelnen VSAM-Datenbank.

- `name`- Gibt den Namen der Datenbankinstanz an, die die Datensätze enthält, die die migrierte Anwendung erstellt.
- `secret-manager-arn`— Gibt den Amazon-Ressourcennamen (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 für migrierte 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"
}
```

user-pool-id

Gibt den Amazon Cognito Cognito-Benutzerpool an, der zur Authentifizierung von Benutzern der migrierten Anwendung AWS Mainframe Modernization verwendet wird. Der AWS-Region Benutzerpool sollte mit dem AWS-Region für die Anwendung übereinstimmen. AWS Mainframe Modernization

Client-ID

Gibt die migrierte Anwendung an, auf die der authentifizierte Benutzer zugreifen kann.

identity-pool-id

Gibt den Amazon Cognito Cognito-Identitätspool an, in dem der authentifizierte Benutzer ein Benutzerpool-Token gegen Anmeldeinformationen eintauscht, die dem Benutzer den Zugriff ermöglichen. AWS Mainframe Modernization Der AWS-Region für den Identitätspool sollte mit dem AWS-Region für die AWS Mainframe Modernization Anwendung übereinstimmen.

## 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 mit Ihrem AD/LDAP details for each AD/LDAP Server einen AWS Secrets Manager geheimen Schlüssel, 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

- Für`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 oder. `awsuser` `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 für den geheimen Schlüssel `Decrypt` und 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"]
}
```

Wenn die Anwendung in LDAP integriert ist und gestartet wurde, müssen Sie Anmeldeinformationen angeben, um mindestens einen anwendungsbezogenen Vorgang ausführen zu können, der in der Liste der unterstützten Autorisierungen aufgeführt ist.

Der LDAP/AD-Authentifizierungshandler ist für Micro Focus (Rocket) 8.0.11 und spätere Versionen verfügbar.

> **Note**
>
> Derzeit muss der LDAP-Administrator in den Unternehmensserverressourcen „OPERCMDS“ in seinem LDAP-Verzeichnis „Alter“ -Berechtigungen für das `casstart` Hilfsprogramm bereitstellen. Dies muss für alle erforderlichen Standardbenutzer (z. B. CICSUSER, wenn die Anwendung CICS-bezogen ist) durchgeführt werden, damit die Anwendung erfolgreich gestartet werden kann.

Um LDAP-Benutzeranmeldedaten für die Authentifizierung und Autorisierung bereitzustellen

1. Erstellen Sie eine AWS Secrets Manager mit den folgenden Schlüsseln und Werten:

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

   > **Important**
   >
   > Sie sollten die Rechte haben, den Secrets Manager auszuführen `DescribeSecrets` und `GetSecretValue` zu verwenden. Ordnen Sie außerdem einen KMS-Schlüssel und die erforderlichen Berechtigungen für den zu AWS Secrets Manager, wie unter Auswählen eines beschrieben AWS KMS key.

2. Wählen Sie den Secrets Manager Manager-Parameter.

AWS console

Bei der Ausführung von Operationen von der AWS Konsole aus besteht die Möglichkeit, den Secrets Manager auszuwählen, der übergeben werden muss.

AWS CLI (or SDK)

Bei der Ausführung von Vorgängen über die AWS-CLI (oder SDK) `auth-secrets-manager-arn` sollte der API-Parameter zusammen mit dem Secrets Manager Manager-ARN übergeben werden.

Hier ist die Liste der Anwendungsvorgänge, die derzeit die Autorisierung unterstützen:

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

## 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": "ADRDSSU"}
    ]
}
```

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. Zum Beispiel:

```
"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.

jcl-file-location

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.

system-procedure-libraries

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 Utilitys. AWS und Batch-Dienstprogramme von Drittanbietern (z. B. M2SFTP, M2WAIT, Syncsort usw.) können optional Aliase haben, sodass die JCL nicht geändert werden muss. Zum Beispiel:

- FDSSORT Alias FDSSORT für SORT und Alias FDSICET für ICETOOL
- ADRDSSU Alias MFADRDSU für ADRDSSU
- Synchronisieren Sie den Sortenalias DMXMFSRT 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"
}
```

binary-file-location

Gibt den Speicherort der CICS-Transaktionsprogrammdateien an.

csd-file-location

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.

system-initialization-table

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.

## Drucker — optional

Geben Sie einen JES-Drucker mit der folgenden Struktur an.

```
"jes-printers": [
    {
        "name": "printerName",
        "classes": [
            "A",
            "B"
        ],
        "description": "printer desc....",
        "exit-module": {
            "name": "lrsprte6",
            "module" : "program"
        }
    }
],
```

> **Note**
>
> Für eine bestimmte Anwendung können maximal 25 Drucker konfiguriert werden.

Name

(Erforderlich) Gibt den Namen an, der dieser Druckerressource zugeordnet werden soll. Namen müssen für jeden Drucker eindeutig sein, und es können maximal 128 alphanumerische Zeichen verwendet werden.

Klassen

(Erforderlich) Gibt die für diese Druckerressource geltenden Ausgabeklassen an. Es können maximal 36 alphanumerische Zeichen verwendet werden.

description

(Optional) Zusätzlicher beschreibender Text für den Drucker.

Modul beenden

(Optional) Gibt ein benutzerdefiniertes Modul für den Druckausgang an. Es gibt keine Standardwerte. Wenn nicht angegeben, wird kein Exit-Modul verwendet. Sie können entweder ein verwaltetes Druck-Exit-Modul verwenden oder ein eigenes bereitstellen. Managed Print-Exit-Module werden unter Verwendung eines reservierten Namens `lrsprte6` für die LRS-Warteschlange definiert oder stellen mithilfe des Modulparameters einen eigenen Namen zur Verfügung, um den Speicherort und den Namen anzugeben.

Die Struktur von `exit-module` besteht aus zwei Komponenten:

- `name`- (Erforderlich), falls `exit-module` verwendet. Der Name des Eintrags des Ausgangsmoduls. Der Name des Eintrags für das Ausgangsmodul ist auf bis zu 8 Zeichen begrenzt.
- `module`— (Optional) Der S3-Speicherort der Binärdatei für das Print-Exit-Modul.

Weitere Beispiele für die Definition des Exit-Moduls finden Sie im the section called “Drucker” Abschnitt.

## XA-Ressourcen — optional

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

> **Note**
>
> Die XA-Ressourcendefinition wurde aktualisiert und enthält nun ein optionales `xa-connection-type` Feld. Falls nicht angegeben, wird davon ausgegangen, dass der Verbindungstyp „Postgres“ ist.

Name

(Erforderlich) Gibt den Namen der XA-Ressource an.

secret-manager-arn

(Erforderlich) Gibt den Amazon-Ressourcennamen (ARN) für das Geheimnis an, das die Anmeldeinformationen für die Verbindung mit der Datenbank enthält.

xa-connection-type

(Optional) Gibt den Verbindungstyp der XA-Ressource an.

Modul

(Erforderlich) Gibt den Speicherort der ausführbaren Datei des RM-Switch-Moduls an. Weitere Informationen finden Sie unter Planung und Design XARs in der Micro Focus Enterprise Server-Dokumentation.

## Laufzeiteinstellungen — optional

Geben Sie die Details an, die für Laufzeiteinstellungen zur Verwaltung zulässiger Umgebungsvariablen erforderlich sind, indem Sie die folgende Struktur verwenden.

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

base-configuration-location

(Optional) Gibt den Speicherort für den Massenimport für eine Micro Focus-Serverkonfiguration an. Diese Datei sollte eine gültige JSON-Datei sein und sich am selben S3-Speicherort wie der oben definierte Speicherort für Anwendungsartefakte befinden. Informationen zum Exportieren der Konfiguration aus einer vorhandenen Anwendung finden Sie im Abschnitt So exportieren Sie eine Region aus ESCWA in der Rocket-Softwaredokumentation.

Umgebungsvariablen

Gibt von Micro Focus unterstützte Umgebungsvariablen an, die auf die Laufzeit dieser Anwendung angewendet werden.

- `ES_JES_RESTART`ist eine Rocket Software-Umgebungsvariable, die die JCL-Neustartverarbeitung ermöglicht. Optional können Sie sie auch `ES_ALLOC_OVERRIDE` als Rocket Software-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 Rocket Software-Umgebungsvariablen im Rocket Enterprise Server for UNIX-Handbuch 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 AWSübertragen. Datensätze müssen in einem Format vorliegen, das in die ausgewählte Runtime-Engine geladen werden kann. In vielen Fällen kann es sich um eine sequentielle Datei handeln, aber für VSAM von Rocket Software (ehemals Micro Focus) muss sie in ihrem eigenen Format vorliegen. Das `DFCONV` Hilfsprogramm ist die empfohlene Methode zur Konvertierung der Datei. Geben Sie den Namen des Buckets und des Ordners in der JSON-Datei mit der Datensatzdefinition an.
> Weitere Informationen zur Runtime Engine von Rocket Software finden Sie unter DFCONV Batch File Conversion in der Rocket Software-Dokumentation.

Weitere Informationen zu AWS Blu Age finden Sie unter. the section called “AWS Blu Age Runtime-Konfiguration”

Themen


- Gemeinsame Eigenschaften
- Beispiel für ein Datensatz-Anforderungsformat für VSAM
- Beispiel für ein Anforderungsformat für einen Datensatz für die GDG-Basis
- Beispiel für ein Anforderungsformat für Datensätze für PS- oder GDG-Generierungen
- Beispiel für ein Anforderungsformat für einen Datensatz 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. (`storageTypeundrelativePath`).
- 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.
- Rocket Software 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 Zielstandort. 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: Nur Datensätze vom Typ VSAM werden unterstützt.
- Runtime-Engine von Rocket Software: 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.
- Rocket Software Runtime Engine: unterstützt.

Wenn Sie VSAM-Datensätze importieren, geben Sie `vsam` als an`datasetOrg`. 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`) und KSDS (). `KS` Das Datensatzformat kann fest oder variabel sein.
- Rocket Software 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

(Erforderlich) 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 und KSDS `min` ist optional und erforderlich. `max`
- Rocket Software 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 Anforderungsformat für einen Datensatz für die GDG-Basis

- AWS Blu Age Runtime Engine: wird nicht unterstützt.
- Rocket Software Runtime Engine: unterstützt.

Wenn Sie GDG-Basisdatensätze importieren, geben Sie `gdg` als an`datasetOrg`. 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-Generierungen

- AWS Blu Age Runtime Engine: wird nicht unterstützt.
- Rocket Software Runtime Engine: unterstützt.

Wenn Sie Datensätze der Generationen PS oder GDG importieren, geben Sie `ps` als and`datasetOrg`. 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 () ?

Datensatzlänge

(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 Anforderungsformat für einen Datensatz für PO

Wenn Sie PO-Datensätze importieren, geben Sie `po` als an`datasetOrg`. Ihr JSON-Code sollte dem folgenden Beispiel ähneln:

```
{
    "storageType": "Database",
    "datasetName": "AWS.M2.PO.PROC",
    "relativePath": "DATA",
```

```
    "datasetOrg": {
        "po": {
            "format": "LSEQ",
            "encoding": "A",
            "memberFileExtensions": ["PRC"]
        }
    },
    "recordLength": {
        "min": 80,
        "max": 80
    }
},
"externalLocation": {
    "s3Location": "s3://$M2_DATA_STORE/source/proc/"
}
}
```

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 Rocket Software 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?
- Für die AWS Mainframe-Modernisierung gerüstet
- Erste Schritte mit der AWS Mainframe-Modernisierung
- Tutorial: Verwaltete Laufzeit für AWS Blu Age einrichten
- Tutorial: Verwaltete Laufzeit für Rocket Software (ehemals Micro Focus) einrichten

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 Rocket Software (ehemals Micro Focus), wenn Sie das Replatforming-Muster verwenden. Sie können auch die Menge an Rechenressourcen auswählen, 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
- Stoppen Sie eine AWS Laufzeitumgebung für die 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-Modernisierungskonsole, um eine Umgebung für die AWS Mainframe-Modernisierung zu erstellen.

Bei diesen Anweisungen wird davon ausgegangen, dass Sie die unter aufgeführten Schritte abgeschlossen haben. Für die AWS Mainframe-Modernisierung gerüstet

## Erstellen Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung zu erstellen

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole 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 Beschreibung der Umgebung 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 (Rocket) 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 EC2 Amazon-Instances, die Ihre Laufzeitumgebung hosten werden. Standalone-Laufzeitumgebungen bieten zwei Optionen für den Instance-Typ und lassen nur eine Instance zu. Laufzeitumgebungen mit hoher Verfügbarkeit bieten zwei Optionen für den Instanztyp und erlauben bis zu zwei Instanzen.

Weitere Informationen finden Sie unter EC2 Amazon-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 den Netzwerktyp aus. Wenn Sie sich dafür entscheiden IPv4, bearbeiten Anwendungen für die AWS Mainframe-Modernisierungsumgebung nur IPv4 Anfragen. Im Dual-Stack-Modus bearbeiten Anwendungen IPv4 sowohl IPv6 Anfragen als auch Anfragen. Wenn Sie den Dual-Stack-Modus wählen, stellen Sie sicher, dass mindestens 1 VPC mit IPv6 -fähigen Subnetzen vorhanden ist.

c. Wählen Sie eine Virtual Private Cloud (VPC).

d. 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 Rocket Software-Engine verwenden, können Sie ein Subnetz angeben.

e. 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 eingehenden und ausgehenden Datenverkehr den Zugriff von einer IP-Adresse auf den Port zulassen, den Sie in der `the section called “Listener – erforderlich”` Eigenschaft der Anwendungsdefinition

angegeben haben. Weitere Informationen finden Sie unter Aktualisieren der Sicherheitsgruppen für Ihren Network Load Balancer im Benutzerhandbuch für Network Load Balancer.

f. Wählen Sie im Feld KMS-Schlüssel die Option Verschlüsselungseinstellungen anpassen aus, wenn Sie vom Kunden verwaltete Verschlüsselungseinstellungen verwenden möchten. AWS KMS key 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

g. (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.

h. Wählen Sie Weiter aus.

7. (Optional) Wählen Sie auf der Seite Speicher anhängen ein oder mehrere Amazon EFS- oder FSx Amazon-Dateisysteme aus.

   Das in einer AWS Mainframe-Modernisierungsumgebung gemountete Dateisystem muss einem geeigneten Benutzer gehören, damit es von Ihren Anwendungen verwendet werden kann, die in der AWS Mainframe Modernization Console ausgeführt werden.

   Um diese Benutzereinstellungen zu konfigurieren, können Sie das Laufwerk an eine EC2 Linux-Amazon-Instance anschließen. Erstellen Sie dann eine Gruppe mit ID `101` und einen Benutzer mit ID`3001`. Stellen Sie außerdem sicher, dass der gewünschte Datenordner, der von Ihren Anwendungen verwendet wird, diesem Benutzer gehören muss.

   Der `myFiles` Ordner kann beispielsweise von Ihren AWS Mainframe-Modernisierungsanwendungen verwendet werden, die in AWS Mainframe Modernization Managed ausgeführt werden.

```
groupadd -g 101 mygroup
useradd -M -g mygroup -p mypassword -u 3001 myuser
```

```
mkdir myFiles
chown myuser:mygroup myFiles
```

> **Note**
>
> Um den Zugriff auf das Dateisystem zu ermöglichen, sollten die folgenden Sicherheitsgruppenregeln für die Einrichtung der Netzwerkkonnektivität zwischen der EFS- und der M2-Umgebungsinstanz konfiguriert werden:
>
> - M2-Umgebungssicherheitsgruppe — Fügen Sie eine ausgehende Regel hinzu, die den Datenverkehr über den NFS 2049-Port zulässt.
> - Sicherheitsgruppe für Dateisystem-Mount-Ziele — Fügen Sie eine eingehende Regel hinzu, die Datenverkehr über den NFS 2049-Port von der Instanz-Sicherheitsgruppe (oben aufgeführt) zulässt, und eine ausgehende Regel, die Datenverkehr über den NFS 2049-Port zulässt.

8. Wählen Sie Weiter aus.
9. Wählen Sie im Abschnitt Wartungsfenster aus, wann ausstehende Änderungen an der Umgebung vorgenommen werden sollen.

   - Wenn Sie „Keine Präferenz“ wählen, wählt AWS Mainframe Modernization ein optimiertes Wartungsfenster für Sie aus.
   - Um ein bestimmtes Wartungsfenster anzugeben, wählen Sie Neues Wartungsfenster auswählen. Wählen Sie dann einen Wochentag, eine Startzeit und eine Dauer für das Wartungsfenster.

   Weitere Informationen zum Wartungsfenster finden Sie unterAWS Wartungsfenster für die Mainframe-Modernisierung.

   Wählen Sie Weiter aus.
10. 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-Modernisierungskonsole, 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 Für die AWS Mainframe-Modernisierung gerüstet voraus.

## Aktualisieren Sie eine Laufzeitumgebung

Um eine Laufzeitumgebung zu aktualisieren

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole 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 angewendet 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 werden und wann 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.

# Stoppen Sie eine AWS Laufzeitumgebung für die Mainframe-Modernisierung

Verwenden Sie die AWS Mainframe-Modernisierungskonsole, 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 Für die AWS Mainframe-Modernisierung gerüstet 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-Modernisierungskonsole, 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.

> **Note**
>
> Sie müssen alle Anwendungen beenden, bevor Sie die Umgebung beenden.

Um eine Laufzeitumgebung zu beenden

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole 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 aus.

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 „Umgebung beenden“.

Um eine Laufzeitumgebung neu zu starten

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole 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 wählen, ob der Neustart sofort erfolgen soll.

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 aus.

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-Modernisierungskonsole, um eine AWS Mainframe-Modernisierungs-Laufzeitumgebung zu löschen.

Diese Anleitungen setzen die Durchführung der Schritte unter Für die AWS Mainframe-Modernisierung gerüstet 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, die Sie löschen möchten, erstellt wurde.
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 Mainframe Modernization Application Testing bietet automatisierte funktionale Äquivalenztests für Ihre Migrationsprojekte. AWS Mainframe Modernization Application Testing beschleunigt Migrationsprojekte, indem es die Elastizität der Cloud nutzt. Sie können unabhängige Testsuiten in beliebig vielen parallel Umgebungen ausführen und so die Testzeiten reduzieren. Zu den wichtigsten Vorteilen von Anwendungstests gehören die Beschleunigung und Agilität der Tests, ein hohes Maß an Wiederholbarkeit der Tests, integrierte Skalierbarkeit und Elastizität, umfangreiche Automatisierung, Kosteneffizienz und nahtlose Integration AWS CloudFormation bei der Erstellung von Zieltestumgebungen.

Themen


- Was sind Anwendungstests für die AWS Mainframe-Modernisierung?
- AWS Konzepte für das Testen von Anwendungen zur Mainframe-Modernisierung
- AWS Voraussetzungen für Anwendungstests zur Mainframe-Modernisierung
- Workflows für Anwendungstestkonsolen
- Tutorial: Richten Sie die CardDemo Beispielanwendung in AWS Mainframe Modernization Application Testing ein
- Tutorial: Wiederholung und Vergleich von Anwendungstests CardDemo für die AWS Mainframe-Modernisierung mithilfe von AWS Blu Age, bereitgestellt auf Amazon EC2
- AWS Mainframe-Modernisierung, Anwendungstests, unterstützte Datensätze, Codepages
- Datenschutz beim Testen von Anwendungen zur AWS Mainframe-Modernisierung
- So funktionieren Anwendungstests für die AWS Mainframe-Modernisierung mit IAM


# Was sind Anwendungstests für die AWS Mainframe-Modernisierung?

Tests wirken sich erheblich auf Modernisierungsprojekte aus. AWS Anwendungstests, eine Funktion von AWS Mainframe Modernization, bieten 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 Testszenarien 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 Konzepte (IaC) genutzt Infrastructure-as-Code werden. Anwendungstests beschleunigen Migrationsprojekte mithilfe der Elastizität der Cloud. Sie können unabhängige Testsuiten 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 die CardDemo Anwendung in Application Testing ein
- the section called “Tutorial: Auf Blu Age erneut abspielen und vergleichen mit AWS 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.

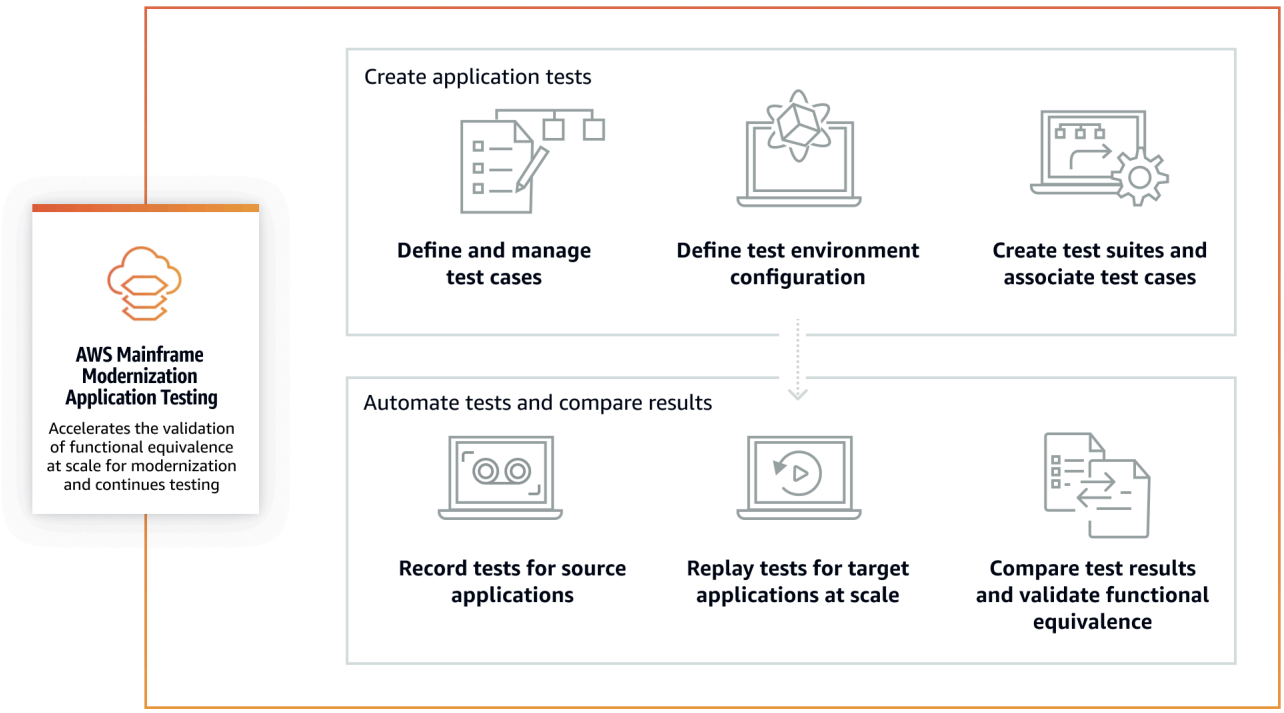
In den AWS CloudFormation Vorlagen, die zum Herunterladen bereitgestellt werden, werden Sie einige allgemeine Funktionen feststellen:

- 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 sorgt für Stabilität, 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 über die Funktionsweise von Anwendungstests.




- Sie können Eingabedaten von der Quelle an die von Ihnen AWS verwendeten Übertragung von Dateien 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-Online-Bildschirme und Benutzerinteraktionen) von Quelle und Ziel. Nachdem Sie Ihr Testszenario auf dem Mainframe ausgeführt haben, erfassen Sie die Ausgabedaten und übertragen sie an AWS das Ziel. Anschließend spielen Sie das Testszenario auf 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. Testfälle erstellen: 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. Definieren Sie die Konfiguration der Testumgebung: Geben Sie Ihre Umgebungskonfiguration an, indem Sie eine AWS CloudFormation Vorlage und zusätzliche Attribute angeben.
3. Testsuiten erstellen: Testsuiten sind eine Sammlung von Testfällen.
4. Laden Sie Datensätze auf der Quelle hoch und spielen Sie sie auf dem Ziel erneut ab: Erfassen Sie die Eingabe- und Ausgabedatensätze auf dem Mainframe und laden Sie sie auf. AWS Spielen Sie das Testszenario anschließend erneut ab. AWS
5. Quell- und Zieldatensätze vergleichen: Application Testing vergleicht automatisch die Ausgabedatensätze von Quell- und Zieldatensätzen, sodass Sie auf einen Blick sehen 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 unter:

- AWS Mainframe-Modernisierung
- AWS CloudFormation

## Zugriff auf Anwendungstests

Sie können auf die Application Testing Console von https://console.aws.amazon.com/apptest/ oder von der AWS Mainframe Modernization Console aus zugreifen, indem Sie im linken Navigationsbereich 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 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
- Testsuite
- Konfiguration der Testumgebung
- Hochladen
- 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
- Daten erneut abspielen
- Referenzdaten
- Hochladen, 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 einer Testsuite 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-3270-Bildschirmtestfälle 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.

## Testsuite

Testsuiten enthalten eine Sammlung von Testfällen, die nacheinander ausgeführt werden. Die Wiedergabe erfolgt auf der Ebene der Testsuite. Alle Testfälle in der Testsuite werden in der Zieltestumgebung ausgeführt, wenn eine Testsuite wiedergegeben wird. Wenn es nach dem Vergleich von Referenz- und Wiederholungstestartefakten Unterschiede gibt, werden die Unterschiede auf Testfallebene angezeigt.

Zum Beispiel Test Suite A:

Testfall 1, Testfall 2, Testfall 3 usw.

## Konfiguration der Testumgebung

Die Konfiguration der Testumgebung ermöglicht es Ihnen, den anfänglichen Satz von Daten und Konfigurationsparametern (oder Ressourcen) einzurichten CloudFormation , die Sie benötigen, um den Testlauf wiederholbar zu machen.

## Hochladen

Uploads werden auf der Ebene der Testsuite durchgeführt. Während des Uploads 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 hochgeladenen Referenzdaten verglichen, um die Gleichwertigkeit der Anwendung sicherzustellen.

## Erneut abspielen

Wiederholungen werden auf Testsuite-Ebene durchgeführt. Während der Wiedergabe verwendet AWS Mainframe Modernization Application Testing das CloudFormation Skript, 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 den Upload einmal auf den Mainframe durchführen und dann mehrmals wiederholen, 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 Upload-Phase hochgeladen und erfasst haben, mit den während der Wiedergabephase generierten Wiedergabedaten verglichen. Vergleiche werden auf der Ebene einzelner Testfälle für Datensätze, Datenbankeinträge und Online-Bildschirme getrennt 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 hochgeladen 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 Testsuiten im Vergleich zu ihren Referenztestsuiten den Status IDENTISCH oder ÄQUIVALENT aufweisen, weist Ihre Testsuite keine Unterschiede auf.

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 Zielsystem + 2 Tage ist, anstatt einen Wert, der genau derselben Spalte beim Referenz-Upload 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 Rocket Software (ehemals Micro Focus) Runtime in läuft. AWS Cloud

## Daten erneut abspielen

Wiedergabedaten werden verwendet, um die Daten zu beschreiben, die durch das Abspielen einer Testsuite in der Zieltestumgebung generiert werden. Wiedergabedaten werden beispielsweise generiert, wenn eine Testsuite 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 behoben 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.

## Hochladen, Wiedergeben und Vergleichen

Das Testen von Anwendungen erfolgt in drei Schritten:

- Upload: erfasst die referenzierten Daten, die auf dem Mainframe für jeden Testfall eines Testszenarios erstellt wurden. Dazu können 3270 Online-Bildschirme, Datensätze und Datenbankeinträge 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 Documentation, um CDC-Journale mit Änderungen zu erfassen und zu generieren.
- Wiederholung: Die Testsuite wird in der Zielumgebung wiedergegeben. Alle in der Testsuite angegebenen Testfälle werden ausgeführt. Spezifizierte Datentypen, die durch die einzelnen Testfälle erstellt wurden, wie Datensätze, relationale Datenbankänderungen oder 3270-Bildschirme, werden automatisiert erfasst. Diese Daten werden als Wiedergabedaten bezeichnet und mit den während der Upload-Phase 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 den Wiederholungsdatensätzen 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 Zielanwendung anzeigt, als Unterschied betrachtet. Ein anderes Beispiel ist ein Upload in einem Datensatz, der zwischen Quell- und Zielanwendungen 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.

## Zielanwendung

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 Zielanwendung zu erreichen. Die Zielanwendung wird normalerweise in der AWS Cloud ausgeführt.

# AWS Voraussetzungen für Anwendungstests zur Mainframe-Modernisierung

AWS Mit der Funktion „Mainframe-Modernization Application Testing“ in AWS Mainframe Modernization können Sie automatisierte funktionale Äquivalenztests für Ihre Migrationsprojekte durchführen. Gehen Sie wie folgt vor, um sich auf die Verwendung der Anwendungstests in der AWS Mainframe-Modernisierungskonsole vorzubereiten:

1. Definieren Sie Testfälle: Definieren Sie die grundlegenden Testeinheiten, die Sie für Ihre Zielanwendung ausführen und in einer bestimmten Reihenfolge wiedergeben möchten. Weitere Informationen zum Erstellen von Testfällen finden Sie unterthe section called “Erstellen Sie Testfälle beim Testen von Anwendungen”.
2. CloudFormation Vorlage und Eingabedaten vorbereiten: Erstellen Sie eine CloudFormation Vorlage, die zur Bereitstellung der Zieltestumgebung verwendet wird. Die Variablen aus dieser Vorlage werden zum Hinzufügen von Eingabedaten und Namen von Ausgabevariablen in Ihrer AWS Mainframe-Modernisierungsanwendung verwendet. Weitere Informationen finden Sie unter Arbeiten mit AWS CloudFormation Vorlagen im AWS CloudFormation Benutzerhandbuch.
3. Stellen Sie den Mainframe-Zugriff und die Datenerfassung sicher: Stellen Sie sicher, dass Sie Zugriff auf den Quell-Mainframe haben. Dadurch wird auch sichergestellt, dass Sie die Quelldaten, die von den auf dem Mainframe ausgeführten Anwendungen generiert wurden, erfassen und hochladen können.

## Workflows für Anwendungstestkonsolen

AWS Die Mainframe Modernization Application Testing Console unterstützt Sie bei der Erstellung von Testfällen, Testsuiten und Testumgebungskonfigurationen.

Themen

- Erstellen Sie Testfälle in AWS Mainframe Modernization Application Testing
- Erstellen Sie Testsuiten in AWS Mainframe Modernization Application Testing
- Erstellen Sie Testumgebungskonfigurationen in AWS Mainframe Modernization Application Testing

# Erstellen Sie Testfälle in AWS Mainframe Modernization Application Testing

Ein Testfall ist eine atomare Einheit, die eine bestimmte Aktion in Ihrem Workflow darstellt. Weitere Informationen zu verschiedenen Konzepten finden Sie unter???.

> **Important**
>
> Sie müssen zuerst mindestens eine Testumgebungskonfiguration erstellen, bevor Sie Testfälle ausführen können. Informationen zum Erstellen Ihrer ersten Umgebungskonfiguration finden Sie unterthe section called “Erstellen Sie Testumgebungskonfigurationen in Application Testing”.

Themen

- Erstellen Sie einen Batch-Testfall
- Erstellen Sie einen Online 3270-Bildschirmtestfall

## Erstellen Sie einen Batch-Testfall

Mit Batch-Testfällen können Sie einen Batch einreichen, um neue Geschäftsdaten (Datenbank- und/oder Datensatzdatensätze) zu lesen, zu verarbeiten und zu ändern oder zu erstellen.

So erstellen Sie einen Batch-Testfall

1. Öffnen Sie die AWS Mainframe Modernization Application Testing Console unterhttps://console.aws.amazon.com/apptest/.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Application Testing verfügbar ist.

   > **Note**
   >
   > Anwendungstests sind derzeit nur in den Regionen USA Ost (Nord-Virginia), Asien-Pazifik (Sydney), Europa (Frankfurt) und Südamerika (São Paulo) verfügbar.

3. Wählen Sie im linken Navigationsbereich Testfälle aus.
4. Geben Sie im Feld Testfall definieren Ihren Testfallnamen und optional eine Beschreibung ein. Wählen Sie unter Testfalltyp die Option Batch aus.
5. Wählen Sie Weiter aus.

6. (Optional) Fügen Sie auf der Seite Batch-JCL-Parameter angeben den JCL-Namen (Job Control Language) und Ihre Job-Parameter (Namen und Werte) hinzu.
7. Wählen Sie Weiter aus.
8. Auf der Seite Zu erfassende Datenquelle können Sie entweder Relationale Datenbankänderungen, Datensätze oder beides auswählen.

   - Wählen Sie Relationale Datenbankänderungen, wenn der Testfall Datenbankeinträge ändern soll.
   - Wählen Sie Datensätze aus, wenn der Testfall Datensätze ändern soll. Fügen Sie unter Ausgabedatensätze den Namen Ihres Ausgabedatensatzes hinzu.

   > **Note**
   >
   > Sie können mehrere Datensätze hinzufügen.

9. Wählen Sie Weiter aus.
10. Überprüfen Sie auf der Seite Überprüfen und erstellen alle Informationen und wählen Sie Testfall erstellen aus.

## Erstellen Sie einen Online 3270-Bildschirmtestfall

Mit Online-3270-Bildschirmtestfällen können Sie interaktive Bildschirmdialoge (3270) ausführen, um neue Geschäftsdaten (Datenbank- und/oder Datensatzdatensätze) zu lesen, zu ändern oder zu erstellen.

Um einen Online-3270-Bildschirmtestfall zu erstellen

1. Öffnen Sie die AWS Mainframe Modernization Application Testing Console unter. https://console.aws.amazon.com/apptest/
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Application Testing verfügbar ist.

   > **Note**
   >
   > Anwendungstests sind derzeit nur in den Regionen USA Ost (Nord-Virginia), Asien-Pazifik (Sydney), Europa (Frankfurt) und Südamerika (São Paulo) verfügbar.

3. Wählen Sie im linken Navigationsbereich Testfälle aus.

4. Geben Sie im Feld Testfall definieren Ihren Testfallnamen und optional eine Beschreibung ein. Wählen Sie unter Testfalltyp die Option Online 3270 screens aus.
5. Wählen Sie Weiter aus.

> **Note**
>
> Für den Online-Bildschirm 3270 müssen Sie keine JCL-Parameter angeben.

6. Wählen Sie Weiter aus.
7. Auf der Seite „Zu erfassende Datenquelle“ ist die Standardauswahl „Online 3270-Bildschirme“. Darüber hinaus können Sie Relationale Datenbankänderungen und Datensätze auswählen.
   - Wählen Sie Relationale Datenbankänderungen, wenn der Testfall Datenbankeinträge ändern soll.
   - Wählen Sie Datensätze aus, wenn der Testfall Datensätze ändern soll. Fügen Sie unter Ausgabedatensätze den Namen Ihres Ausgabedatensatzes hinzu.

> **Note**
>
> Sie können mehrere Datensätze hinzufügen.

8. Wählen Sie Weiter aus.
9. Überprüfen Sie auf der Seite Überprüfen und erstellen alle Informationen und wählen Sie Testfall erstellen aus.

## Erstellen Sie Testsuiten in AWS Mainframe Modernization Application Testing

Testsuiten sind eine Reihe von Testfällen, die in einer sequentiellen Reihenfolge ausgeführt werden. Testsuiten sind wichtig für die Wiedergabe von Testfällen.

> **Important**
>
> Bevor Sie Testsuiten erstellen können, benötigen Sie mindestens einen Testfall. Sie können Ihren ersten Testfall mit, erstellenthe section called “Erstellen Sie Testfälle beim Testen von Anwendungen”.

Weitere Informationen zu verschiedenen Konzepten finden Sie unterthe section called “Konzepte für das Testen von Anwendungen”.

Themen

- Erstellen Sie eine Testsuite
- Laden Sie Referenzdaten hoch
- Wiedergeben und vergleichen

## Erstellen Sie eine Testsuite

Mit Testsuiten können Sie verschiedene Testfälle ausführen und sie später erneut abspielen und vergleichen.

Um eine Testsuite zu erstellen

1. Öffnen Sie die AWS Mainframe Modernization Application Testing Console unterhttps://console.aws.amazon.com/apptest/.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Application Testing verfügbar ist.

   > **Note**
   >
   > Anwendungstests sind derzeit nur in den Regionen USA Ost (Nord-Virginia), Asien-Pazifik (Sydney), Europa (Frankfurt) und Südamerika (São Paulo) verfügbar.

3. Wählen Sie im linken Navigationsbereich Testfälle aus.
4. Wählen Sie Testsuiten erstellen aus.
5. Suchen Sie im Abschnitt Testsuiten erstellen nach Testfällen aus der Testfallbibliothek und wählen Sie Ausgewählte Testfälle hinzufügen aus.

   > **Note**
   >
   > Sie können einer Testsuite bis zu 20 Testfälle hinzufügen.

6. Geben Sie im Bereich Testsuite den Namen Ihrer Testsuite und optional eine Beschreibung ein. Wählen Sie außerdem entweder eine verwaltete oder eine nicht verwaltete Laufzeit aus. Dadurch wird definiert, wie die Testsuite eine AWS Mainframe-Modernisierungsanwendung

konfiguriert und dekonfiguriert. Fügen Sie optional den JSON S3-URI für den AWS Mainframe-Modernisierungs-Importdatensatz hinzu.

7. Stapeln Sie im Abschnitt Hinzugefügte Testfälle Ihre Testfälle in der Reihenfolge, in der Sie sie hochladen und erneut abspielen möchten.
8. Wählen Sie Testsuite erstellen aus.

## Laden Sie Referenzdaten hoch

Laden Sie Mainframe-Referenzdaten in AWS Application Testing hoch. Sie müssen die hochgeladenen Referenzdaten nur beim ersten Mal speichern. Der Testservice kann die hochgeladenen Ergebnisse aus der Quelle wiederverwenden und sie nacheinander mit den wiedergegebenen Ergebnissen auf dem Ziel vergleichen.

Um Referenzdaten hochzuladen

1. Wählen Sie im Abschnitt Testsuiten die Testsuite aus, um Referenzdaten hochzuladen.
2. Klicken Sie auf Upload.
3. Wählen Sie auf der Seite Referenzdaten hochladen die Testfälle aus, die Sie erneut abspielen möchten. Füllen Sie die Felder für Datum der Datenerfassung, Speicherort des Journals für Datenbankänderungen (S3) und Speicherort der Datensätze (S3) aus und wählen Sie Hochladen aus.

## Wiedergeben und vergleichen

Der Vorgang „Wiedergeben und Vergleichen“ ordnet Ihren Testfall der Zieltestumgebung zu und führt die Anwendung aus. Sie müssen Daten hochladen, bevor Sie den Wiedergabevorgang ausführen können.

Zum Abspielen und Vergleichen

1. Wählen Sie im Abschnitt Testsuiten die Testsuite aus, die Sie erneut abspielen möchten.
2. Wählen Sie „Wiedergeben und vergleichen“.
3. Wählen Sie auf der Übersichtsseite „Wiedergeben und vergleichen“ Ihre Testumgebungskonfiguration aus und überprüfen Sie die Informationen. Mit der Bearbeitungsfunktion können Sie alle Felder für die Konfiguration der Testumgebung bearbeiten. Sie können auch AWS CloudFormation Parameter finden.

4. Wählen Sie im Abschnitt Testfälle, die wiedergegeben werden sollen, die Testfälle aus und ordnen Sie sie in der Reihenfolge an, in der Sie sie wiedergeben möchten.
5. Wählen Sie „Wiedergeben und vergleichen“.

# Erstellen Sie Testumgebungskonfigurationen in AWS Mainframe Modernization Application Testing

Testumgebungskonfigurationen ermöglichen es Ihnen, den anfänglichen Satz von Daten und Konfigurationsparametern (oder Ressourcen) einzurichten AWS CloudFormation , die Sie benötigen, um den Testlauf wiederholbar zu machen.

Weitere Informationen zu verschiedenen Konzepten finden Sie unterthe section called “Konzepte für das Testen von Anwendungen”.

## Erstellen Sie eine Konfiguration für die Testumgebung

Konfigurieren Sie Ihre Testumgebung so, dass Testfälle in Application Testing wiedergegeben und verglichen werden.

Richten Sie Konfigurationen der Testumgebung ein

1. Öffnen Sie die AWS Mainframe Modernization Application Testing Console unterhttps://console.aws.amazon.com/apptest/.
2. Wählen Sie in der AWS-Region Auswahl die Region aus, in der Application Testing verfügbar ist.

   > **Note**
   >
   > Anwendungstests sind derzeit nur in den Regionen USA Ost (Nord-Virginia), Asien-Pazifik (Sydney), Europa (Frankfurt) und Südamerika (São Paulo) verfügbar.

3. Wählen Sie im linken Navigationsbereich Testumgebungskonfigurationen aus.
4. Wählen Sie Testumgebungskonfiguration erstellen aus.
5. Geben Sie im Bereich Testumgebungskonfiguration erstellen den Namen und die Beschreibung ein. Fügen Sie auch Ihren Amazon S3 S3-Bucket hinzu, der die CloudFormation Vorlage für Anwendungstests enthält. Darüber hinaus können Sie die CloudFormation Eingabeparameter hinzufügen, die bei der CloudFormation Stack-Erstellung verwendet werden.

6. Geben Sie Ihre AWS-Mainframe-Modernisierungsanwendung an, die von dieser Testkonfiguration betroffen sein wird. Fügen Sie den Namen der Ausgabevariablen für die Anwendungs-ID der AWS Mainframe-Modernisierung und die Runtime-Engine (AWS Blu Age nicht verwaltet oder von Rocket Software (früher Micro Focus) verwaltet) hinzu.

   > **Note**
   >
   > Der Name der Ausgabevariablen für die Anwendungs-ID von AWS Mainframe Modernization sollte mit dem Namen der Ausgabevariablen aus der CloudFormation Vorlage für die Stack-Erstellung übereinstimmen.

   > **Important**
   >
   > Bei der nicht verwalteten Laufzeit von AWS Blu Age müssen Sie außerdem den Namen der Ausgabevariablen für die VPC-Endpunkt-Service-ID, den Namen der Ausgabevariablen für den Listener-Port und den Namen der Ausgabevariablen für den Namen angeben. WebApp Diese Namen sollten mit den Namen der Ausgabevariablen aus der Vorlage übereinstimmen. CloudFormation

7. (Optional) Zusätzliche Attribute wie der Name der Ausgabevariablen können für die Aufgabe Amazon Resource Name (ARN) des Database Migration Service (DMS) definiert werden, die für die Erfassung relationaler Datenbankänderungen verwendet wird. Ein weiteres Attribut ist der DDL S3-URI der Quelldatenbank.

   > **Important**
   >
   > Der Name der Ausgabevariablen sollte mit dem Variablennamen aus der CloudFormation Vorlage übereinstimmen.

8. (Optional) Passen Sie Ihren Key Management Service (KMS) -Schlüssel an. Weitere Informationen finden Sie unter Verwalten des Zugriffs auf kundenverwaltete Schlüssel im Entwicklerhandbuch zum AWS Key Management Service .
9. Wählen Sie Testumgebungskonfiguration erstellen aus.

# Tutorial: Richten Sie die CardDemo Beispielanwendung in AWS Mainframe Modernization Application Testing ein

In diesem Tutorial erstellen Sie einen AWS CloudFormation Stack, der Sie bei der Einrichtung der CardDemo Beispielanwendung für das Replatforming mit Micro Focus on AWS Mainframe Modernization Managed Service und Funktionen wie AWS Mainframe Modernization Application Testing unterstützt. Das Tutorial beschreibt eine AWS CloudFormation Beispielvorlage, mit der Sie den Stack erstellen können. Wir stellen auch eine komprimierte Datei mit den erforderlichen Anwendungsartefakten zur Verfügung. Die Beispielvorlage 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. Beispiel, `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 Testsuiten 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. Beispiel, `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 vor `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
```

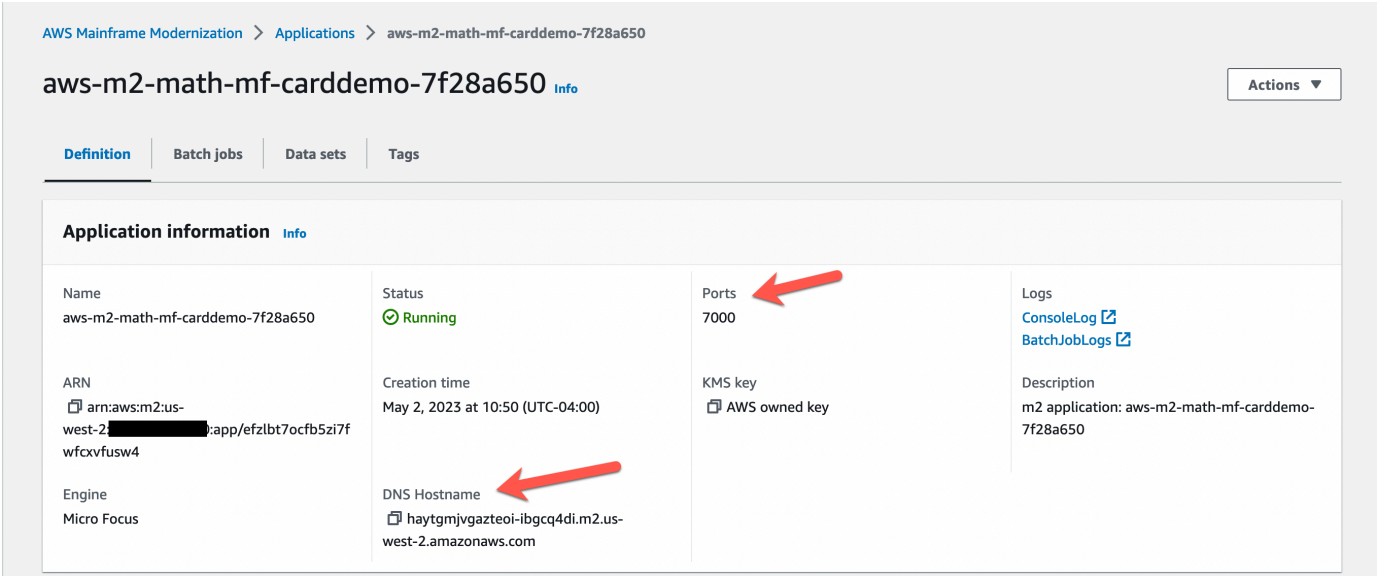
port

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.




# Tutorial: Wiederholung und Vergleich von Anwendungstests CardDemo für die AWS Mainframe-Modernisierung mithilfe von AWS Blu Age, bereitgestellt auf Amazon EC2

In diesem Tutorial führen Sie die erforderlichen Schritte aus, um Test-Workloads mit der auf Amazon bereitgestellten CardDemo Anwendung abzuspielen und zu vergleichen, die auf AWS Blu Age ausgeführt wird. EC2

## Schritt 1: Besorgen Sie sich AWS Blu Age EC2 Amazon Amazon Machine Image (AMI)

Folgen Sie den Anweisungen im Setup-Tutorial für AWS Blu Age Runtime (bei 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 EC2 Amazon-Instance mit dem AWS Blu Age AMI

1. Richten Sie Ihre AWS Anmeldeinformationen 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 EC2 Amazon-Instance erstellt haben, finden Sie im EC2 Amazon-Image-Katalog die richtige AMI-ID, für die das AMI aus dem Amazon S3-Bucket wiederhergestellt wurde.

> **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 >= 3 und die CLI-Version muss >= 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 mit dem Namen amzn-s3-demo-bucket in dieser Dokumentation kopiert werden.

## Schritt 4: Datenbanken laden und die Anwendung initialisieren CardDemo

Erstellen Sie eine temporäre EC2 Amazon-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 EC2 Amazon-Instance zu erstellen, die zum Generieren der Datenbank-Snapshots verwendet wird.

1. Generieren Sie Ihr EC2 key pair, das für die EC2 Instanz verwendet wird, und geben Sie es an. 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://amzn-s3-demo-bucket/databases'
   ```

3. Geben Sie den Amazon S3 S3-Pfad Ihres Ordners an, in den Sie den Dateisystemordner aus dem vorherigen Schritt abgelegt haben:

   ```
   S3ApplicationFilesPath:
       Description: 'S3 application files folder path'
       Type: String
   ```

```
    Default: 's3://amzn-s3-demo-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://amzn-s3-demo-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.

```
S3SaveProducedFilesPath:
    Description: 'S3 path folder to save produced files'
    Type: String
    Default: 's3://amzn-s3-demo-bucket/post-produced-files'
```

6. Ändern Sie die AMI-ID mithilfe der folgenden Vorlage durch die richtige, die Sie zuvor 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, die beim Ausführen der Ladedatenbanken erstellt werden, mit CloudFormation ändern. 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 habenSaveProducedFilesPath, überprüfen, ob die Datensätze korrekt erstellt wurden.

## Schritt 5: Starten Sie AWS Blu Age Runtime CloudFormation

Wird verwendet CloudFormation , um die EC2 Amazon-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 beim Start des CFN 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 EC2 Amazon-Schlüsselpaar für die EC2 Instance mit diesem Befehl an:

```
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 BaaAmiId, 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:

```
S3ApplicationFilesPath:
    Description: 'bucket name'
    Type: String
    Default: 's3://amzn-s3-demo-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://amzn-s3-demo-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:

Zum Beispiel:

```
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 ACMCertArn ändern:

```
ACMCertArn:
    Type: String
    Description: 'ACM certificate ARN'
    Default: 'your arn certificate'
```

- (Optional) Die Standardauthentifizierung ist standardmäßig deaktiviert, und der Parameter ist auf false WithBacBasicAuthenticationgesetzt. 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 EC2 Amazon-Instanz

Führen Sie die CloudFormation Vorlage manuell aus, um die AWS Blu Age EC2 Amazon-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 EC2 Amazon-Zielinstanz während der Wiedergabe und des Vergleichs automatisch zu erstellen.

1. Führen CloudFormation Sie den Befehl create stack aus, um die AWS Blu Age EC2 Amazon-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 EC2 Amazon-Instance suchen. Stellen Sie mit dem Sitzungsmanager eine Connect.
3. Nachdem Sie mit der EC2 Amazon-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: Überprüfen Sie, ob 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 EC2 Amazon-Instance erfolgreich über die AWS CloudFormation Vorlage erstellt.
2. Der Tomcat-Service auf dem AWS Blu Age bei Amazon EC2 ist ausnahmslos in Betrieb.

Wenn Sie die EC2 Instance mit der CardDemo Anwendung zum Laufen gebracht haben, führen Sie die folgenden Schritte in der Application-Testing-Konsole durch, um die Wiedergabe und den Vergleich von Batch-Datensätzen durchzuführen.

## Schritt 8: 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 `CREASTMT.JCL` als JCL-Namen an.
3. Fügen Sie der Testfalldefinition die folgenden Datensätze hinzu:

| Name | CCSID | RecordFormat | RecordLength |
|---|---|---|---|
| AWS.M2.CARDDEMO.STATEMNT.PS | „037" | FB | 80 |
| AWS.M2.CARDDEMO.STATEMNT.HTML | „037" | FB | 100 |

> **Note**
>
> Ihr JCL-Name und die Datensatzdetails müssen übereinstimmen.

## Schritt 9: Erstellen Sie eine Testsuite

1. Erstellen Sie eine neue Testsuite und geben Sie einen Namen und eine Beschreibung dafür ein.
2. Fügen Sie den Testfall, den Sie im vorherigen Schritt erstellt haben, zu Ihrer Testsuite hinzu.
3. Sobald die Testsuite erstellt ist, erfassen Sie die Testfälle auf dem Mainframe und laden Sie die Mainframe-Referenzdaten in AWS Application Testing hoch.
4. Wählen Sie Testsuite erstellen aus.

## Schritt 10: Erstellen Sie eine Testumgebungskonfiguration

1. Erstellen Sie eine neue Testumgebungskonfiguration und geben Sie einen Namen und eine Beschreibung dafür ein.
2. Fügen Sie Ihre CloudFormation Vorlage hinzu. Sie können auch den Namen und den Wert des Eingabeparameters aus Ihrer CloudFormation Vorlage hinzufügen.
3. Wählen Sie den AWS Mainframe Modernization Service AWS Blu Age non-managed als Laufzeit.
4. Fügen Sie den Namen der Ausgabevariablen als Namen für die Anwendungs-ID der AWS Mainframe-Modernisierung, den Namen der Ausgabevariablen für die Dienst-ID des VPC-Endpunkts, den Namen der Ausgabevariablen für den Listener-Port und den Namen der Ausgabevariablen als Namen hinzu. WebApp

   > **Note**
   >
   > Die Namen dieser Felder sollten mit den Namen der Ausgabevariablen aus der CloudFormation Vorlage übereinstimmen, die bei der Stack-Erstellung von AWS Mainframe Modernization zurückgegeben wird.

5. (Optional) Wählen Sie den Namen der Ausgabevariablen für den ARN der DMS-Aufgabe (Database Migration Service) und den S3-URI-Speicherort der Quelldatenbank DDL (Database Definition Language).
6. (Optional) Passen Sie Ihren Key Management Service (KMS) -Schlüssel an. Weitere Informationen finden Sie unter Verwalten des Zugriffs auf kundenverwaltete Schlüssel im Entwicklerhandbuch zum AWS Key Management Service .
7. Wählen Sie Testumgebungskonfiguration erstellen aus.

## Schritt 11: Laden Sie Ihre Eingabedaten in die Testsuite hoch

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 der Testsuite die Schaltfläche Hochladen aus.
4. Geben Sie auf der Seite Referenzdaten hochladen den Amazon S3 S3-Speicherort an, an den Sie die vom Quell-Mainframe abgerufenen Datensätze hochgeladen haben.
5. Wählen Sie Upload, um den Upload-Vorgang 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 die Testsuite und die Testfälle in der EC2 Zielumgebung von AWS AWS Blu Age on Amazon 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 Replay und vergleichen Sie. Es sollte ungefähr drei Minuten dauern, den CloudFormation Stack zu erstellen und den Vergleich durchzuführen.

Sobald alles abgeschlossen ist, sollten Sie Vergleichsergebnisse mit einigen Unterschieden haben, die absichtlich für diese Demo erstellt wurden.

# AWS Mainframe-Modernisierung, Anwendungstests, unterstützte Datensätze, Codepages

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, Brasilien, Australien, Neuseeland |
| 273 | IBM273, IBM-273, Cp 273 | Gastgeber: Österreich, Deutschland |
| 277 | IBM277, IBM-277, Cp277 | Gastgeber: Dänemark, Norwegen |
| 278 | IBM278, IBM-278, CP278 | Gastgeber: Finnland, Schweden |
| 280 | IBM280, IBM-280, Cp280 | Gastgeber: Italien |
| 284 | IBM284, IBM-284, CP284 | Gastgeber: Spanien, Lateinamerika (Spanisch) |
| 285 | IBM285, IBM-285, CP285 | Gastgeber: Vereinigtes Königreich |
| 297 | IBM297, IBM-297, CP297 | Gastgeber: Frankreich |
| 300 | IBM-300 | JAPAN DB EBCDIC |
| 301 | IBM-301 | PC-Daten: Japan DB |
| 437 | IBM437, 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, Internati onal Latin-1 |
| 720 | IBM-720 | MADOS ARABISCH |
| 737 | IBM-737, x- IBM737 | MADOS GRIECHISCH |

| CCSID | Zeichensätze | Beschreibung |
|---|---|---|
| 775 | IBM775, IBM-775 | MADOS BALTISCH |
| 808 | IBM-808 | PC-Daten: Kyrillisch, Russland, mit Euro |
| 813 | ISO-8859-7, _7 ISO8859 | ISO 8859-7: Griechenland |
| 819 | ISO-8859-1_1 ISO8859 | ISO 8859-1: Latin-1-Länder |
| 833 | IBM-833 | KOREANISCH EBDDISCH |
| 834 | IBM-834, x- IBM834 | KOREANISCHER DB EBCDIC |
| 835 | IBM-835 | T-CHINESISCHES DB EBCD |
| 836 | IBM-836 | S-CHINESISCH EBCDISCH |
| 837 | IBM-837 | S-CHINESISCH EBCDISCH |
| 850 | IBM850, IBM-850, CP850 | PC-Daten: Latin-1-Länder |
| 855 | IBM855, IBM-855, Cp 855 | PC-Daten: Kyrillisch |
| 856 | IBM-856, x-, Cp 856 IBM856 | 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 | IBM860, IBM-860 | PC-Daten: Portugiesisch |
| 861 | IBM861, IBM-861 | PC-Daten: Island |
| 862 | IBM862, IBM-862, Cp 862 | PC-Daten: Hebräisch (Migration) |
| 863 | IBM863, IBM-863 | PC-Daten: Kanada |

| CCSID | Zeichensätze | Beschreibung |
|---|---|---|
| 865 | IBM865, IBM-865, Cp 865 | PC-Daten: Den/Norwegen |
| 866 | IBM866, IBM-866, Cp 866 | PC-Daten: Kyrillisch, Russland |
| 867 | IBM-867 | PC-Daten: Hebräisch mit Euro |
| 870 | IBM870, IBM-870, CP870 | Gastgeber: Latin-2, mehrsprachig |
| 871 | IBM871, IBM-871, Cp 871 | Gastgeber: Island |
| 874 | x- IBM874 | PC-Daten: Thailändisch |
| 875 | IBM-875, x-, Cp875 IBM875 | Gastgeber: Griechenland |
| 897 | IBM-897 | PC-Daten: Japan SB |
| 912 | ISO-8859-2, _2 ISO8859 | ISO 8859-2: Latin-2 mehrsprachig |
| 915 | ISO-8859-5_5 ISO8859 | ISO 8859-5: Kyrillisch |
| 916 | ISO-8859-8, _8 ISO8859 | ISO 8859-8: Hebräisch |
| 918 | IBM918, IBM-918, CP918 | Gastgeber: Urdu |
| 920 | ISO-8859-9_9 ISO8859 | ISO 8859-9: Latin-5 (ECMA-128, Türkei TS-5881) |
| 921 | IBM-921, x-, CP921 IBM921 | PC-Daten: Lettland, Litauen |
| 922 | IBM-922, x-, Cp922 IBM922 | PC-Daten: Estland |
| 923 | ISO-8859-15, Cp923, _15_FDIS ISO8859 | 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, IBM93 x-0, CP930 | Katakana-Host: erweitertes SBCS. Kanji-Host: DBCS mit 4370 benutzerdefinierten Zeichen |
| 932 | IBM-932 | PC-Daten: Japan Mix |
| 933 | IBM-933, x-, Cp933 IBM933 | Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerd efinierten Zeichen und 11172 vollständigen Hangul-Zeichen |
| 935 | IBM-935, x-, CP935 IBM935 | Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerd efinierten Zeichen. |
| 937 | IBM-937, x-, CP937 IBM937 | Host: Erweitertes SBCS. Host: DBCS mit 6204 benutzerd efinierten Zeichen |
| 939 | IBM-939, x-, CP939 IBM939 | Latin Host: erweitertes SBCS. Kanji-Host: DBCS mit 4370 benutzerdefinierten Zeichen. |
| 942 | IBM-942, IBM-942C, x-IBM942, x-C, CP942, CP942C IBM942 | PC-Daten: Erweitertes SBCS. PC-Daten: DBCS mit 1880 benutzerdefinierten Zeichen |
| 943 | IBM-943, IBM-943C, Shift_JIS , Fenster-31j, Fenster-932, x-, x-C, Cp943, Cp943C, IBM943 IBM943 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-, CP948 IBM948 | PC-Daten: Erweitertes SBCS. PC-Daten: DBCS mit 6204 benutzerdefinierten Zeichen |
| 949 | IBM-949, IBM-949C, x-IBM949, x-C, CP949, CP949C IBM949 | IBM KS Code - PC-Daten: SBCS. IBM KS-Code — PC-Daten: DBCS mit 1880 benutzerdefinierten Zeichen |
| 950 | Big5, IBM-950, x-0, Cp950 IBM95 | 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-, CP964 IBM964 | G0: ASCII. G1: CNS 11643 Ebene 1. G1: CNS 11643 Flugzeug 2. |
| 970 | EUC-KR, x-0, Cp 970 IBM97 | G0: ASCII. G1: KSC X5601-1989 einschließlich 1880 benutzerdefinierter Zeichen |
| 971 | IBM-971 | KOREANISCHE EUC |
| 1006 | IBM-1006, x-006, Cp1006 IBM1 | ISO-8: Urdu |
| 1025 | IBM-1025, x-025, CP1025 IBM1 | Gastgeber: Kyrillisch, mehrsprachig |

| CCSID | Zeichensätze | Beschreibung |
|---|---|---|
| 1026 | IBM1026, IBM-1026, Cp1026 | Gastgeber: Latin-5 (Türkei) |
| 1027 | IBM-1027 | JAPANISCHES LATEINISCHES EBCD |
| 1041 | IBM-1041 | PC-Daten: Japan |
| 1043 | IBM-1043 | PC-Daten: T-Chinesisch |
| 1046 | IBM-1046, IBM-1046S, x-046 IBM1 | 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, _6 ISO8859 | ISO 8859-6: Arabisch |
| 1097 | IBM-1097, x-097, CP1097 IBM1 | Gastgeber: Farsi |
| 1098 | IBM-1098, x-098, CP1098 IBM1 | PC-Daten: Farsi |
| 1112 | IBM-1112, x-, Cp1112 IBM1112 | Gastgeber: Lettland, Litauen |
| 1114 | IBM-1114 | PC-Daten: T-CH SB |
| 1115 | IBM-1115 | PC-Daten: S-CH GB |
| 1122 | IBM-1122, x-, Cp1122 IBM1122 | Gastgeber: Estland |
| 1123 | IBM-1123, x-, Cp1123 IBM1123 | Gastgeber: Cyrillic Ukraine |

| CCSID | Zeichensätze | Beschreibung |
|---|---|---|
| 1124 | IBM-1124, x-, Cp1124 IBM1124 | 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: Vereinigtes Königreich, 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, Internati onal 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, UTF8 | 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 Koran DBCS, einschließlich 11172 vollständiger Hangul-Version |
| 1364 | IBM-1364 | Host: Erweitertes SBCS. Host: DBCS mit 1880 benutzerdefinierten Zeichen und 11172 vollständigen Hangul-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-, Cp1381 IBM1381 | 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, IBM-1383 GB2312, x-, Cp1383 IBM1383 | 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-Hos t: DBCS mit 6205 benutzerd efinierten 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 030, 2-Byte-Daten GB18 030, 4-Byte-Daten GB18 |
| 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 | IBMeucJP |

# Datenschutz beim Testen von Anwendungen zur AWS Mainframe-Modernisierung

Das Modell der AWS gemeinsamen Verantwortung gilt für den Datenschutz in AWS Mainframe Modernization Application Testing. Wie in diesem Modell beschrieben, AWS ist es verantwortlich für den Schutz der globalen Infrastruktur, 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-Beitrag AWS -Modell der geteilten Verantwortung und in der DSGVO im AWS -Sicherheitsblog.

Wir empfehlen, dass Sie AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. Dadurch

erhält jeder Benutzer nur die Berechtigungen, die für die Erfüllung 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 AWS Ressourcen zu kommunizieren. 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 Ihnen, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden in Tags oder frei formatierten Textfeldern (z. B. dem Feld Name) zu verwenden. Dies gilt auch, wenn Sie mit AWS Mainframe Modernization Application Testing oder anderen Anwendungen arbeiten und AWS-Services dabei die Konsole, die API oder verwenden. AWS CLI AWS SDKs Alle Daten, die Sie in Tags oder freie Textfelder für Namen eingeben, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie eine URL für einen externen Server angeben, vermeiden Sie es, Anmeldeinformationen in der URL zu verwenden, um Ihre Anfrage an diesen Server zu überprüfen.

## Daten, die im Rahmen von AWS Mainframe Modernization Application Testing gesammelt wurden

AWS Mainframe Modernization Application Testing sammelt verschiedene Arten von Daten von Ihnen:

- `Resource definition`: Die Ressourcendefinition gibt die Daten an, die an Application Testing übergeben werden, wenn Sie eine Ressource vom Typ Testfall, Testsuite oder Testkonfiguration erstellen oder aktualisieren.
- `Scripts for replay`: Dies sind Skripts, die an Application Testing für Ihre AWS Mainframe-Modernisierungsanwendung übergeben wurden.

- `Data for comparison`: Dies sind Datensätze oder CDC-Dateien (Database Change Data Capture), die zum Vergleich an Application Testing übergeben wurden.

AWS Mainframe Modernization Application Testing speichert diese Daten nativ in. AWS Die Daten, die wir von Ihnen sammeln, werden in einem von AWS Mainframe Modernization Application Testing verwalteten Amazon S3 S3-Bucket gespeichert. Wenn Sie eine Ressource löschen, werden die zugehörigen Daten aus dem Amazon S3 S3-Bucket entfernt.

Wenn Sie einen Testlauf starten, um eine Wiedergabe zum Testen interaktiver Workloads durchzuführen, lädt AWS Mainframe Modernization Application Testing das Skript in einen flüchtigen, von Amazon ECS verwalteten Fargate-Container herunter, um die Wiedergabe durchzuführen. Die Skriptdatei wird gelöscht, sobald die Wiedergabe abgeschlossen ist und die skriptgenerierte Ausgabedatei in einem von Application Testing verwalteten Amazon S3 S3-Bucket in Ihrem Konto gespeichert wird. Die Replay-Ausgabedatei wird aus dem Amazon S3 S3-Bucket gelöscht, wenn Sie den Testlauf löschen.

Wenn Sie einen Testlauf starten, um Dateien (Datensätze oder Datenbankänderungen) zu vergleichen, lädt AWS Mainframe Modernization Application Testing die Dateien in einen flüchtigen, von Amazon ECS verwalteten Fargate-Container herunter, um den Vergleich durchzuführen. Die heruntergeladenen Dateien werden gelöscht, sobald der Vergleichsvorgang abgeschlossen ist. Die Vergleichsausgabedaten werden in Ihrem Konto in einem von Application Testing verwalteten Amazon S3 S3-Bucket gespeichert. Die Ausgabedaten werden aus dem S3-Bucket gelöscht, wenn Sie den Testlauf löschen.

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 Application Testing zum Vergleichen von Dateien verwendet.

## Datenverschlüsselung im Ruhezustand für den AWS Mainframe Modernization Application Testing

AWS Mainframe Modernization Application Testing ist in AWS Key Management Service (KMS) integriert, um eine transparente serverseitige Verschlüsselung (SSE) für alle abhängigen Ressourcen bereitzustellen, die Daten dauerhaft speichern. Zu den Ressourcenbeispielen gehören Amazon Simple Storage Service, Amazon DynamoDB und Amazon Elastic Block Store. AWS Mainframe Modernization Application Testing 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 können Sie damit Anwendungen testen, für die strenge Verschlüsselungsvorschriften und gesetzliche Anforderungen gelten.

Sie können diese Verschlüsselungsebene nicht deaktivieren oder einen anderen Verschlüsselungstyp auswählen, wenn Sie Testfälle, Testsuiten oder Testkonfigurationen erstellen.

Sie können Ihren eigenen, vom Kunden verwalteten Schlüssel für Vergleichsdateien und AWS CloudFormation Vorlagen verwenden, um Amazon S3 zu verschlüsseln. Sie können diesen Schlüssel verwenden, um alle Ressourcen zu verschlüsseln, die für Testläufe in Application Testing erstellt wurden.

> **Note**
>
> DynamoDB-Ressourcen werden immer mit einem Dienstkonto Von AWS verwalteter Schlüssel im Application Testing verschlüsselt. Sie können DynamoDB-Ressourcen nicht mit einem vom Kunden verwalteten Schlüssel verschlüsseln.

AWS Mainframe Modernization Application Testing verwendet Ihren vom Kunden verwalteten Schlüssel für die folgenden Aufgaben:

- Exportieren von Datensätzen aus Application Testing nach Amazon S3.
- Hochladen von Vergleichsausgabedateien auf Amazon S3.

Weitere Informationen finden Sie unter Kundenverwaltete Schlüssel im AWS Key Management Service Entwicklerhandbuch.

## Einen kundenverwalteten Schlüssel erstellen

Sie können einen symmetrischen, vom Kunden verwalteten Schlüssel erstellen, indem Sie den AWS Management Console oder den verwenden. AWS KMS APIs

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.

Im Folgenden finden Sie ein Beispiel für eine Schlüsselrichtlinie mit eingeschränktem Zugriffsbereich ViaService , die es Application Testing ermöglicht, Daten, die durch Wiederholung und Vergleich generiert wurden, in Ihr Konto zu schreiben. Sie sollten diese Richtlinie der IAM-Rolle zuordnen, wenn Sie die API aufrufen. `StartTestRun`

Example

```
{
    "Sid": "TestRunKmsPolicy",
    "Action": ["kms:Decrypt", "kms:GenerateDataKey"],
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/TestRunRole"
    },
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "kms:ViaService": ["s3.amazonaws.com"]
        },
        "ForAnyValue:StringEquals": {
            "kms:EncryptionContextKeys": "aws:apptest:testrun"
        }
    }
}
```

Weitere Informationen finden Sie unter Verwalten des Zugriffs auf kundenverwaltete Schlüssel im Entwicklerhandbuch zum AWS Key Management Service .

Weitere Informationen zur Fehlerbehebung beim Schlüsselzugriff finden Sie im AWS Key Management Service -Entwicklerhandbuch.

# Angabe eines vom Kunden verwalteten Schlüssels für das Testen von AWS Mainframe-Modernisierungsanwendungen

Wenn Sie eine Testkonfiguration erstellen, können Sie einen vom Kunden verwalteten Schlüssel angeben, indem Sie eine KEY-ID eingeben. Application Testing verwendet die Verschlüsselung der Daten, die während des Testlaufs in den Amazon S3 S3-Bucket hochgeladen wurden.

- SCHLÜSSEL-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.

Um Ihren vom Kunden verwalteten Schlüssel hinzuzufügen, wenn Sie eine Testkonfiguration mit dem erstellen AWS CLI, geben Sie den `kmsKeyId` Parameter wie folgt an:

```
create-test-configuration --name test \
--resources '[{
    "name": "TestApplication",
    "type": {
        "m2ManagedApplication": {
            "applicationId": "wqju4m2dcz3rhny5fpdozrsdd4",
            "runtime": "MicroFocus"
        }
    }
}]' \
--service-settings '{
    "kmsKeyId": "arn:aws:kms:us-west-2:111122223333:key/05d467z6-c42d-40ad-b4b7-274e68b14013"
}'
```

# AWS Verschlüsselungskontext für Mainframe-Modernisierung, Anwendungstests

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 Mainframe-Modernisierung, Anwendungstests

AWS Mainframe Modernization Application Testing verwendet denselben Verschlüsselungskontext für alle AWS KMS kryptografischen Operationen im Zusammenhang mit einem Testlauf, wobei der Schlüssel `aws:apptest:testrun` und der Wert die eindeutige Kennung des Testlaufs sind.

Example

```
"encryptionContext": {
        "aws:apptest:testrun": "u3qd7uhdandgdkhhi44qv77iwq"
}
```

Verwenden des Verschlüsselungskontexts für die Überwachung

Wenn Sie einen symmetrischen, vom Kunden verwalteten Schlüssel zur Verschlüsselung Ihres Testlaufs verwenden, können Sie den Verschlüsselungskontext auch in Prüfaufzeichnungen und Protokollen verwenden, um festzustellen, wie der vom Kunden verwaltete Schlüssel beim Hochladen von Daten auf Amazon S3 verwendet wird.

## Überwachung Ihrer Verschlüsselungsschlüssel für das Testen von Anwendungen zur AWS Mainframe-Modernisierung

Wenn Sie einen vom AWS KMS Kunden verwalteten Schlüssel mit Ihren AWS Mainframe Modernization Application Testing-Ressourcen verwenden AWS CloudTrail, können Sie damit Anfragen verfolgen, die AWS Mainframe Modernization Application Testing beim Hochladen von Objekten an Amazon S3 sendet.

## Verschlüsselung während der Übertragung

Bei Testfällen, die Schritte zum Testen von Transaktionsworkloads definieren, erfolgt der Datenaustausch zwischen dem verwalteten Terminalemulator von Application Testing, auf dem Ihre Selenium-Skripte ausgeführt werden, und den Anwendungsendpunkten von AWS Mainframe Modernization während der Übertragung nicht verschlüsselt. AWS Beim Testen von Mainframe-Modernisierungsanwendungen wird eine Verbindung AWS PrivateLink zu Ihrem Anwendungsendpunkt hergestellt, um Daten privat auszutauschen, ohne dass der Datenverkehr über das öffentliche Internet offengelegt wird.

AWS Mainframe Modernization Application Testing verwendet HTTPS, um den Service zu verschlüsseln. APIs Die gesamte andere Kommunikation innerhalb von AWS Mainframe

Modernization Application Testing wird durch die Service-VPC oder Sicherheitsgruppe sowie durch HTTPS geschützt.

Die Standardverschlüsselung bei der Übertragung ist standardmäßig konfiguriert, gilt jedoch nicht für `TN3270` protokollbasierte interaktive Workload-Tests.

# So funktionieren Anwendungstests für die AWS Mainframe-Modernisierung mit IAM

Bevor Sie IAM verwenden, um den Zugriff auf AWS Mainframe Modernization Application Testing zu verwalten, sollten Sie sich darüber informieren, welche IAM-Funktionen für AWS Mainframe Modernization Application Testing verfügbar sind.

IAM-Funktionen, die Sie mit Mainframe Modernization Application Testing verwenden können AWS

| IAM-Feature | AWS Unterstützung für Anwendungstests bei der Mainframe-Modernisierung |
|---|---|
| Identitätsbasierte Richtlinien | Ja |
| Ressourcenbasierte Richtlinien | Nein |
| Richtlinienaktionen | Ja |
| Richtlinienressourcen | Ja |
| Bedingungsschlüssel für die Richtlinie | Ja |
| ACLs | Nein |
| ABAC (Tags in Richtlinien) | Ja |
| Temporäre Anmeldeinformationen | Ja |
| Forward Access Sessions (FAS) | Ja |
| Servicerollen | Nein |
| Serviceverknüpfte Rollen | Nein |

Einen allgemeinen Überblick darüber, wie Anwendungstests und andere AWS Services für die AWS Mainframe-Modernisierung mit den meisten IAM-Funktionen funktionieren, finden Sie im AWS IAM-Benutzerhandbuch unter Services, die mit IAM funktionieren.

## Identitätsbasierte Richtlinien für Anwendungstests zur 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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 Anwendungstests zur Mainframe-Modernisierung AWS

Beispiele für identitätsbasierte Richtlinien für AWS Mainframe-Modernisierungs-Anwendungstests finden Sie unter. Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

## Ressourcenbasierte Richtlinien im Rahmen von Mainframe Modernization Application Testing 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 Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

## Politische Maßnahmen für Anwendungstests zur AWS Mainframe-Modernisierung

Unterstützt Richtlinienaktionen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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 Modernization Application Testing-Aktionen finden Sie in der Service Authorization Reference unter Actions defined by AWS Mainframe Modernization Application Testing.

Bei Richtlinienaktionen in AWS Mainframe Modernization Application Testing wird vor der Aktion das folgende Präfix verwendet:

```
apptest
```

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie mit Kommata:

```
"Action": [
      "apptest:CreateTestCase",
      "apptest:StartTestRun"
]
```

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": "apptest:List*"
```

Beispiele für identitätsbasierte Richtlinien für die AWS Mainframe-Modernisierung finden Sie unter. Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

## Richtlinienressourcen für Anwendungstests zur AWS Mainframe-Modernisierung

Unterstützt Richtlinienressourcen: Ja

Administratoren können mithilfe von AWS JSON-Richtlinien angeben, wer auf was Zugriff hat. Das heißt, welcher Prinzipal Aktionen für welche Ressourcen und 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 AWS Mainframe Modernization Application Testing-Ressourcen einschränken, indem Sie sie verwenden ARNs , um die Ressource zu identifizieren, für die die IAM-Richtlinie gilt. Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) in der Allgemeine AWS-Referenz.

Ein Testfall für AWS Mainframe Modernization Application Testing hat beispielsweise den folgenden ARN.

```
"Resource": "arn:aws:apptest:regionId:accountId:testcase/service-generated-unique-identifier"
```

Eine Testkonfiguration für AWS Mainframe Modernization Application Testing hat den folgenden ARN.

```
"Resource": "arn:aws:apptest:regionId:accountId:testconfiguration/service-generated-unique-identifier"
```

Eine AWS Mainframe Modernization Application Testing Testsuite hat den folgenden ARN.

```
"Resource": "arn:aws:apptest:regionId:accountId:testsuite/service-generated-unique-identifier"
```

Ein Testlauf für AWS Mainframe Modernization Application Testing hat den folgenden ARN.

```
"Resource": "arn:aws:apptest:regionId:accountId:testrun/service-generated-unique-identifier"
```

Nicht alle AWS Mainframe Modernization Application Testing-Aktionen unterstützen Berechtigungen auf Ressourcenebene. Für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, müssen Sie den Platzhalter (*) verwenden.

Die folgenden AWS Mainframe Modernization Application Testing-Aktionen unterstützen keine Berechtigungen auf Ressourcenebene.

```
ListTestCases
ListTestConfigurations
ListTestRuns
ListTestSuites
ListTagsForResource
```

Eine Liste der Ressourcentypen und der AWS zugehörigen Ressourcentypen für Mainframe Modernization Application Testing finden Sie in der Service Authorization Reference unter Resources defined by AWS Mainframe Modernization Application Testing. ARNs Informationen darüber, mit welchen Aktionen Sie den ARN jeder Ressource angeben können, finden Sie unter Von AWS Mainframe Modernization Application Testing definierte Aktionen.

Beispiele für identitätsbasierte Richtlinien zur AWS Mainframe-Modernisierung finden Sie unter. Beispiele für identitätsbasierte Richtlinien für die Mainframe-Modernisierung AWS

## Richtlinien-Bedingungsschlüssel für Anwendungstests zur 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, AWS 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.

Eine Liste der Bedingungsschlüssel für AWS Mainframe Modernization Application Testing finden Sie unter Bedingungsschlüssel für AWS Mainframe Modernization Application Testing in der Service

Authorization Reference. Informationen zu den Aktionen und Ressourcen, mit denen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von AWS Mainframe Modernization Application Testing 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) beim Testen von AWS Mainframe-Modernisierungsanwendungen

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

## Attributbasierte Zugriffskontrolle (ABAC) mit Anwendungstests für die 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 Bedingungselement 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 Definieren von Berechtigungen mit ABAC-Autorisierung im IAM-Benutzerhandbuch. Um ein Tutorial mit Schritten zur Einstellung von ABAC anzuzeigen, siehe Attributbasierte Zugriffskontrolle (ABAC) verwenden im IAM-Benutzerhandbuch.

## Temporäre Anmeldeinformationen beim Testen von AWS Mainframe-Modernisierungsanwendungen 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 von einer Benutzerrolle zu einer IAM-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.

## Leiten Sie Zugriffssitzungen für AWS Anwendungstests zur Mainframe-Modernisierung weiter

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 Application Testing ohne Ihre ausdrückliche Zustimmung Zugriff auf Kundendaten. AWS Mainframe Modernization Application Testing beispielsweise liefert Testergebnisse an einen Amazon S3 S3-Bucket eines Kunden, ohne die ausdrückliche Genehmigung des Kunden einzuholen. Möglicherweise müssen Sie alle Compliance-Unterlagen entsprechend aktualisieren.

## Servicerollen für das Testen von Anwendungen zur AWS Mainframe-Modernisierung

Unterstützt Servicerollen: Nein

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.

## Servicebezogene Rollen für Anwendungstests im Zusammenhang mit der AWS 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.

# 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 bietet verschiedene Laufzeitoptionen für unterschiedliche Einsatzszenarien und Betriebspräferenzen. Dazu gehören verwaltete und nicht verwaltete Laufzeiten mit jeweils eigenen Funktionen und Bereitstellungszielen.

Einen umfassenden Überblick über die verfügbaren AWS Blu Age Runtime-Optionen, einschließlich verwalteter und nicht verwalteter Versionen, Bereitstellungsziele und ihrer jeweiligen Merkmale, finden Sie in der Dokumentation. the section called “AWS Optionen für Blu Age Runtime”

Dieser Leitfaden hilft Ihnen dabei, die Unterschiede zwischen den Runtime-Optionen zu verstehen und die für Ihr Modernisierungsprojekt am besten geeignete auszuwählen.

Themen

- AWS Veröffentlichungen von Blu Age
- AWS Konzepte von Blu Age Runtime
- Konfiguration für AWS Blu Age Runtime einrichten
- AWS Blu Age Runtime APIs
- AWS Blu Age Runtime einrichten (nicht verwaltet)
- Ändern Sie den Quellcode mit Blu Age Developer IDE
- AWS Blu Age Häufig gestellte Fragen


# AWS Veröffentlichungen von Blu Age

AWS Die Blu Age-Engine hat mehrere Versionen, aus denen Sie wählen können. Diese Seite bietet einen Überblick darüber, wie die AWS Blu Age-Versionierung funktioniert, welche Änderungen die einzelnen Versionsversionen haben, Anleitungen zur Aktualisierung von Versionen, wie Updates für AWS Blu Age mit Kunden kommuniziert werden und den Lebenszyklus dieser Versionen.

the section called “AWS Blu-Age-Versionierung”Auf der Seite finden Sie ausführliche Informationen zu Veröffentlichungen und dazu, wie jede Version anhand von Haupt- und Nebenversionen identifiziert werden kann. Die the section called “AWS Versionshinweise zu Blu Age” Seite enthält ausführliche Versionshinweise für jede Haupt- und Nebenversion. the section called “AWS Sicherheitslücken in Blu Age”Auf dieser Seite wird erwähnt, wie AWS Blu Age mit Common Vulnerabilities and Exposures (CVE) umgeht. the section called “ AWS Blu Age aktualisieren”enthält Anweisungen zum Upgrade für AWS Blu Age-Versionen. Und the section called “AWS Lebenszyklus von Blu Age” enthält alle Informationen zu den Enddaten (EOL) der Hauptversionen von AWS Blu Age Runtime.

Themen


- AWS Blu-Age-Versionierung
- AWS Optionen für Blu Age Runtime
- AWS Versionshinweise zu Blu Age
- AWS Sicherheitslücken in Blu Age
- Upgrade-Anweisungen für AWS Blu Age
- AWS Lebenszyklus von Blu Age

# AWS Blu-Age-Versionierung

Die Produkte AWS Blu Age Transformation und Runtime werden mithilfe eines Semver-konformen Schemas (Semantic Versioning) versioniert. Um Ihre Anwendung bereitzustellen, müssen Sie die entsprechende Runtime-Version verwenden, die mit Ihrem modernisierten Code kompatibel ist. Wenn Sie Fragen dazu haben, welche Version Sie verwenden sollen, wenden Sie sich an Ihren AWS Blu Age-Liefermanager.

## Versionen

Jede Version ist mit einem **`[Major].[Minor].[Patch]`**Muster gekennzeichnet. Bei der AWS Blu Age Runtime-Version `4.1.0` ist die Hauptversion beispielsweise 4, die Nebenversion ist 1 und die Patch-Version ist 0.

Wir beabsichtigen, monatlich neue Nebenversionen von AWS Blu Age Runtime und neue Hauptversionen zu veröffentlichen, wenn es wichtige Änderungen am Produkt oder seinen Abhängigkeiten gibt.

Einzelheiten zu den neuen Funktionen, die in den einzelnen Versionen verfügbar sind, finden Sie unter. the section called “AWS Versionshinweise zu Blu Age”

## Alpha-Vorabversionen

Jede Alpha-Vorabversion ist mit einem **`[Major].[Next_Minor].0-alpha.[pre-release]`**Muster gekennzeichnet. In der Vorabversion `4.2.0-alpha.1` `alpha.1` werden die in verfügbaren Änderungen beispielsweise in der nächsten Nebenversion veröffentlicht. `4.2.0`

Alpha-Vorabversionen sind häufig kurzlebige Versionen, die für eine schnelle Iteration während der Modernisierungsprojekte vorgesehen und verfügbar sind. Für neue Alpha-Vorabversionen gibt es keinen festen Veröffentlichungsrhythmus. Sie werden verfügbar gemacht, sobald sie entwickelt und getestet werden.

Weitere Informationen zu Versionierung, Upgrades und Support finden Sie unter. Lebenszyklus der Komponenten

> **Important**
>
> Alpha-Vorabversionen sollten nur während der Phase des Modernisierungsprojekts und nicht für produktive oder kritische Workloads verwendet werden.

# AWS Optionen für Blu Age Runtime

AWS Blu Age bietet drei Arten von Runtime-Optionen, um den verschiedenen Phasen Ihrer Modernisierung und Ihren betrieblichen Anforderungen gerecht zu werden. Auf dieser Seite werden die einzelnen Optionen, ihre Eigenschaften, Anwendungsfälle und der Zugriff auf sie beschrieben.

## Nicht verwaltete Runtime

Mit AWS Blu Age Runtime (nicht verwaltet) können Sie Ihre modernisierte Anwendung in Ihrer eigenen Umgebung bereitstellen AWS-Konto und so Ihre eigene Infrastruktur verwalten. Diese Option bietet sowohl Release-Versionen als auch Vorabversionen, sodass Sie alle technischen Komponenten, die erforderlich sind, um Ihre modernisierte Anwendung nach Ihren Wünschen auszuführen, flexibel einsetzen können. Sie können zwischen stabilen Versionen für Produktionsumgebungen oder Vorabversionen für Test- und Entwicklungszwecke wählen.

Die nicht verwaltete Runtime wird vom Kunden bereitgestellt und verwaltet und bietet so mehr Kontrolle über die Laufzeitumgebung. Sie bietet automatisierte Refactoring-Funktionen und eignet sich für maßgeschneiderte Bereitstellungsszenarien.

Wann sollte dies verwendet werden?

Die nicht verwaltete Laufzeit eignet sich für Test- und Produktionsumgebungen und ist besonders nützlich, wenn eine spezifische Anpassung der Laufzeitumgebung erforderlich ist.

So greift man zu

Informationen zum Anfordern des Zugriffs auf die nicht verwalteten Runtime-Artefakte finden Sie unter Onboarding von AWS Blu Age Runtime.

Bereitstellung

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

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

Die Bereitstellung auf Amazon EC2 kann direkt in der Instance oder über eine containerisierte Docker-Anwendung erfolgen. Dies ist die bevorzugte Methode bei der Verwendung von Amazon ECS oder Amazon EKS.

Ausführliche Anweisungen zur Bereitstellung finden Sie in der Dokumentation zur Einrichtung von AWS Blu Age Runtime (nicht verwaltet).

### Verwaltete Laufzeit

Mit AWS Blu Age Runtime Managed können Sie Ihre modernisierte Anwendung in einer AWS-verwalteten Umgebung bereitstellen, die Ihre Erfahrung vereinfacht, sodass Sie die zugrunde liegende Infrastruktur, auf der Ihre modernisierte Anwendung ausgeführt wird, nicht verwalten müssen.

Die verwaltete Runtime bietet eine von ihr verwaltete Infrastruktur mit automatischen AWS Updates und Patches. Sie bietet einen vereinfachten Betrieb und eine vereinfachte Wartung. In der verwalteten Runtime sind nur Release-Versionen verfügbar, wodurch Stabilität und Zuverlässigkeit für Ihre Produktionsumgebungen gewährleistet sind.

Wann sollte dies verwendet werden?

Die verwaltete Runtime ist ideal für Produktionsumgebungen und eignet sich am besten für Situationen, in denen Sie bei der Laufzeitverwaltung einen unkomplizierten Ansatz bevorzugen.

So greift man zu

Um auf die AWS Blu Age Runtime Managed Version zuzugreifen, benötigen Sie lediglich Zugriff auf die folgenden AWS Dienste:

- Amazon S3
- AWS Mainframe-Modernisierung

Mit den entsprechenden Kontoberechtigungen können Sie nahtlos mit der verwalteten Laufzeitumgebung interagieren. Dieser optimierte Zugriff stellt sicher, dass Sie den vollen Funktionsumfang von Blu Age Runtime nutzen und gleichzeitig von den Managed Services AWS profitieren können.

Bereitstellung

Informationen zur Einrichtung und Verwendung von Managed Runtime finden Sie in der Dokumentation Managed Runtime für AWS Blu Age einrichten.

## Developer Runtime (BluInsights Toolbox)

Die Developer Runtime ist über die BluInsights Toolbox zugänglich. Sie wurde für Entwicklungs- und Testphasen konzipiert und wird regelmäßig mit den neuesten Funktionen aktualisiert. Diese Runtime umfasst sowohl Release-Versionen als auch Alpha-Vorabversionen und bietet Entwicklern Zugriff auf stabile Builds sowie modernste Funktionen. Ihr Hauptzweck ist die Unterstützung bestimmter Test- oder Entwicklungsaktivitäten in einer lokalen Entwicklungsumgebung, in der Regel von einer IDE aus. Wichtig ist, dass diese Laufzeit auf 2 Stunden begrenzt ist und sich daher eher für gezielte Entwicklungssitzungen als für langfristige Bereitstellungen oder Produktionsbereitstellungen eignet.

Wann sollte dies verwendet werden?

Die Developer Runtime eignet sich ideal für erste Entwicklungs- und Modernisierungsprojekte sowie für schnelle Iterationen und Tests modernisierter Anwendungen.

So greift man zu

Der Zugriff auf die BluInsights Toolbox erfolgt im Rahmen Ihres AWS Blu Age-Projekts. Die Developer Runtime ist über Anfragen zur AWS Blu Age-Toolbox verfügbar. Nach der Genehmigung haben Sie Zugriff auf bestimmte S3-Buckets: `s3://toolbox-dev-runtime-<region>`

Diese Buckets sind in den Regionen US-East-1 und US-East-2 verfügbar. Um den verfügbaren Bucket zu verwenden, fügen Sie die Region an den Bucket-Namen an (z. B.). `s3://toolbox-dev-runtime-us-east-1`

Detaillierte Anweisungen, wie Sie Zugriff beantragen und die erforderlichen Berechtigungen einrichten können, finden Sie in der Dokumentation Dev and Special AWS Blu Age Runtimes.

Bereitstellung

So stellen Sie die Developer Runtime bereit:

1. Verfügbare Versionen auflisten mit AWS CLI:

```
aws s3 ls s3://toolbox-dev-runtime
```

2. Wählen Sie eine bestimmte Version aus und listen Sie deren Inhalt auf:

```
aws s3 ls s3://toolbox-dev-runtime/[version]/
```

3. Laden Sie das Runtime-Artefakt herunter:

```
aws s3 cp s3://toolbox-dev-runtime/[version]/gapwalk-[version]-dev.tar.gz
```

4. Extrahieren Sie die Runtime und richten Sie sie entsprechend Ihren Projektanforderungen ein.

Ausführlichere Bereitstellungsanweisungen und Nutzungsrichtlinien finden Sie in der Dokumentation Dev and Special AWS Blu Age Runtimes.

> **Note**
>
> Stellen Sie sicher, dass Sie über die erforderlichen S3-Leseberechtigungen für AWS-Konto den Zugriff auf diese Buckets verfügen. Ein Beispiel für eine IAM-Richtlinie finden Sie in der Dokumentation Dev and Special AWS Blu Age Runtimes.

## AWS Versionshinweise zu Blu Age

Dieser Abschnitt enthält die Versionshinweise von 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 4.8.0
- Runtime-Version 4.8.0
- AWS Blu Age Transformationsengine 4.8.0
- Versionshinweise 4.7.0
- Runtime-Version 4.7.0
- AWS Blu Age Transformationsengine 4.7.0
- Versionshinweise 4.6.0
- Runtime-Version 4.6.0

- AWS Blu Age Transformationsengine 4.6.0
- Versionshinweise 4.5.0
- Runtime-Version 4.5.0
- AWS Blu Age Transformation Engine 4.5.0
- Versionshinweise 4.4.0
- Runtime-Version 4.4.0
- AWS Blu Age Transformation Engine 4.4.0
- Versionshinweise 4.3.0
- Runtime-Version 4.3.0
- Version 4.3.0 der Modernisierungstools
- Versionshinweise 4.2.0
- Runtime-Version 4.2.0
- Version 4.2.0 der Modernisierungstools
- 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 4.8.0

Veröffentlichungsdatum: 23. April 2025

Diese Version von AWS Blu Age Runtime and Transformation Engines enthält wichtige Updates zur Verbesserung der Funktionen und Leistung der Datenbanktransformation:

- IBM IMS-Datenbanktransformation — Zusätzliche Unterstützung für die Transformation von IBM IMS-Datenbanken in unsere maßgeschneiderte JHDB-Lösung Blu Age.
- In-Memory-Cache-Funktion — Es wurde eine In-Memory-Cache-Funktion hinzugefügt, die es Benutzern ermöglicht, schreibgeschützte Daten im Speicher zwischenzuspeichern, wodurch die Leistung datenintensiver Programme verbessert wird.

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 22.11.0 |
| | Npm 10.9.0 |
| | Eckig 19.1.3 |
| Serviceschicht | Spring Boot 3.4.2 |
| | Spring Core 6.2.2 |

| | |
|---|---|
| | Frühjahrssitzung 3.4.1 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 15.10 |
| | Oracle 21c |
| Anwendungsserver | Apache Tomcat 10.1.17 |

## Runtime-Version 4.8.0

zOS

Neue Features

- Unterstützung für die Definition benutzerdefinierter Header für gesicherte Transaktionen über die YML-Eigenschaft eingeführt. `gapwalk-application.security.customAllowedHeaders` Diese Eigenschaft bezieht sich auf eine `gapwalk-application.identity` Eigenschaft mit Wert. `oauth`
- Es wurde eine Funktion hinzugefügt, mit der Kunden Blusam-Metadaten auf der Grundlage der tatsächlichen Datensätze in der Blusam-Datenbanktabelle neu erstellen können
- Gesicherte JHDB-Verbindungen mithilfe von Integration AWS Secrets Manager

Verbesserungen

- COBOL
  - Verbesserte Unterstützung für die AT-END-Bedingung in der READ-Dateianweisung
  - Verbesserte Unterstützung für MOVE signierter numerischer Literale in alphanumerische Felder, ohne das Zeichen zu verschieben
  - Verbesserte Unterstützung für die UNSTRING-Anweisung zur Verarbeitung von Konvertierungen nicht numerischer in numerische Felder und zur Beibehaltung niedriger Werte, wenn das Trennzeichen nicht gefunden und kein Wert in andere Variablen verschoben wird
  - Die Adressbindungsoperation für dynamische Linkage-Entitäten wurde verbessert
  - Verbesserte Operationen zur Verkettung von Zeichenketten durch die Unterstützung von Shift In/Out-Bytes (SOSI)

  - Verbesserte Unterstützung für die COBOL ACCEPT-Anweisung mit Quellcode DAY-OF-WEEK
  - Unterstützung für SYSIN und SYSOUT als SPECIAL NAMES-Werte hinzugefügt
  - Verbesserte Unterstützung für die STRING-Verkettung zur Verarbeitung des Formular-Feed-Zeichens
- JCL - IDCAMS
  - Unterstützung für das Löschen von GDG-Generationen mit einem Basisnamen ohne den GDG-Parameter hinzugefügt
  - Verbesserte Unterstützung für die Behandlung des VOLUME-Parameters als Variante des Standardparameters VOLUMES für die Anweisung DEFINE CLUSTER
- JCL - SORTIEREN
  - Unterstützung für SYMNAMES mit dem Befehl JOINKEYS wurde hinzugefügt
- JCL - BIEGENER
  - Verbesserte Unterstützung für IEBGENER beim Umgang mit ungültigen SYSIN-Inhalten durch Einführung von Validierungsprüfungen und automatischem Fallback auf ICEGENER, wodurch ein kontinuierlicher Betrieb auch bei falschen Steueranweisungen gewährleistet wird
- JCL — INFUTILB
  - Die Behandlung der UNLOAD-INTO- und FROM-Klauseln wurde verbessert
- JCL - DSNUTILB
  - Unterstützung für den DOUBLE PRECISION-Datentyp hinzugefügt
- JCL - Verschiedenes
  - Das Überschreiben von Dateikonfigurationen mithilfe von DD-Namen in der untergeordneten Prozedur des Jobs wurde verbessert
  - DSNTIAUL/INFUTILB — Verbesserte Anpassung des SYSREC-Padding-Bytes für das Unload-Hilfsprogramm
  - Unterstützung für die OUTDD-Option für die DISPLAY-Anweisungsumleitung hinzugefügt
  - Verbesserte Unterstützung für das Schreiben von Fixed Block Sequential File. Wenn sich die LRECL im COBOL-Programm von der JCL-LRECL unterscheidet, hat die JCL-LRECL Vorrang.
  - Unterstützung für Katalog-DCB-Attribute für die SORTOUT-Datei hinzugefügt, jedoch nur für permanente Datensätze
  - Verbesserte Unterstützung für die bedingte JCL-Schritttransformation
  - Verbesserte Unterstützung für ACCEPT mit SYSIN und Zeilen, die die Zielgröße überschreiten, mit abschließenden Leerzeichen

  - Verbesserte Unterstützung für SET-Anweisungen zur Bearbeitung von Befehlen ohne Variablennamen. Der darin definierte Befehl beinhaltet eine Mitgliedsdatei, auf die innerhalb der JCL zugegriffen werden kann, und der Befehl enthält Inline-Kommentare
  - Es wurde Unterstützung hinzugefügt, um die ScriptContext-Parameter auf Jobebene in den JCL-Checkpoint-Metadaten und den Job-Kontext für die Verwendung beim Neustart beizubehalten
- Blusam
  - Das Einfügen von Datensätzen bei großen indizierten Blusam-Datensätzen beim Schreiben nach dem Laden eines leeren Datensatzes wurde verbessert
  - Verbesserte Leistung bei großen Datensätzen durch Aktivierung eines Aufwärmmechanismus und Einführung eines optionalen Prefetch-Fensters für Datensätze sowie lokaler Speicher für Indizes und Seiten
  - Verbesserte Unterstützung für Export Data Set zur Verarbeitung eines größeren Datensatzes mit AWS Key Management Service
- SQL
  - Verbesserte Unterstützung für den JDBC BLOB-Datentyp
- Integration der Nachrichtenwarteschlange
  - Unterstützung für User Identifier im MQ PUT-Befehl hinzugefügt, wenn XA aktiviert ist
  - Unterstützung für gleichzeitige MQ-Verbindungen hinzugefügt, wenn XA aktiviert ist
- IMS
  - Es wurde eine YML-Eigenschaft hinzugefügt`jhdb.keepParent`, um zu steuern, ob IMS-ISRT-Aufrufe die Felder _parent und _logicalparent in der Datenbank auffüllen, wenn diese Eigenschaft auf true gesetzt ist
  - Die Stabilität wurde verbessert, da die Nachrichtenwarteschlange für ungültigen Transaktionscode entfernt wurde

AS400

Neue Features

- Es wurde eine In-Memory-Cache-Funktion hinzugefügt, die es Benutzern ermöglicht, schreibgeschützte Daten im Speicher zwischenzuspeichern, wodurch die Leistung für datenintensive Programme verbessert wird.

Verbesserungen

- CL
  - Verbesserte Unterstützung des QRYSLT-Parameters im OPNQRYF-Befehl zur Verarbeitung von RANGE-Schlüsselwörtern, zur Analyse von Werten, wenn Zeichenkettenspalten rein numerische Werte enthalten, und zur Verarbeitung leerer Zeichenkettenwerte
  - Unterstützung für den FTP-Befehl SENDPASV wurde hinzugefügt
  - Verbesserte Unterstützung für die Abfragebildung im Befehl DSPDBR, der mit Qcmdexc ausgeführt wurde
  - Die Unterstützung der MONMSG-Ausnahme wurde verbessert, damit sie ordnungsgemäß im Ausführungsstapel weitergegeben wird
  - Das Kopieren von Datensätzen zwischen Tabellen mit unterschiedlichen Spaltenzahlen und Datentypen auf CPYF wurde verbessert
  - Verbesserte Priorität bei der Nachrichtenüberwachung: Nachrichten auf Befehlsebene haben jetzt wie vorgesehen Vorrang vor Nachrichten auf Prozedurebene
  - Verbesserte Unterstützung für den CPYF-Befehl, um die von IVLIB definierte Bibliothek zu berücksichtigen BLU4
  - Verbesserte Unterstützung für den Befehl CHKOBJ, um den Fall zu behandeln, dass die Datei nicht MonMsgs in der Bibliothek existiert
- Screen
  - Die Behandlung von BigDecimal Werten mit mehr als 15 Ziffern wurde verbessert, indem der Wert manuell erstellt und dann in eine Zahl umgewandelt wurde, die angezeigt werden soll, anstatt die Zahl zu runden
- Misc (Verschiedenes)
  - Verbesserte Unterstützung für die Option „ UserSpace Ersetzen“ als NEIN, wenn diese Option UserSpace bereits vorhanden ist
  - Verbesserte BLU4 IVLIB-Funktionalität, die sowohl das Abrufen als auch das Schreiben von Datensätzen in überschriebene Bibliotheken unterstützt
  - Verbesserte Unterstützung für mehrdimensionale Array-Felder mithilfe von DIM- und OCCURS-Operationen
  - Verbesserte Unterstützung für niedrige Werte, die im DAO-Betrieb als Schlüsselfeldwerte übergeben werden
  - Der Konflikt bei der MONMSG-Anweisung für Nachrichten, die von einem lokalen und einem globalen Monitor verarbeitet werden können, wurde behoben
  - Die Einstellung für das Einfügen FileStatus doppelter Datensätze wurde verbessert

- Der Konflikt mit der Sperre des Benutzerbereichs wurde reduziert und das Risiko eines Fehlers bei der Erstellung wurde reduziert, indem zuerst die Registrierung verwendet wurde, um zu überprüfen, ob ein Benutzerbereich vorhanden ist
- Verbesserte Unterstützung für das Überspringen von Datensätzen nach delete readPriorEqual ()
- Verbesserter Mechanismus zum Sperren von Datensätzen unter Commitment-Kontrolle
- Verbesserte Handhabung der RPG-Anweisung RETURN in Subroutinen
- Die Unterstützung von Benutzerinformationen wurde verbessert in SharedContext

## AWS Blu Age Transformationsengine 4.8.0

zOS

Verbesserungen

- COBOL
  - Verbesserte Generierung des Druckermodus für mehrere Dateien.
  - Verbesserte Unterstützung für SPECIAL-NAMES zur Verarbeitung optionaler Endpunkte und mehrerer Punkte.
  - Verbesserte Unterstützung von COPY mit relativen -REPLACING-Anweisungenpath /REPLACE OFF/COPY.
  - Unterstützung für COL als gültiger Feldname hinzugefügt.
  - Verbessertes Parsen zur Behandlung von Leerzeichen nach SPECIAL-NAMES, X-COMPUTER und DECLARATIVES.
  - Unterstützung für S01 bis S05 und C01 bis C12 als gültige Absatznamen hinzugefügt.
  - Unterstützung für die LENG-Funktion als Alias der LENGTH-Funktion hinzugefügt.
  - Das Parsen von Copybooks, wenn der Wert der cobolMaxCode Width-Konfiguration vom COBOL-Includer abweicht, wurde verbessert
  - Verbesserte Unterstützung für das Parsen von Ausdrücken mit Komma und Operation, z. B. MIN (A * B, (3 + 1)/2, 3 + 4).
  - Unterstützung für IS INITIAL PROGRAM in der PROGRAM ID-Klausel hinzugefügt.
  - Unterstützung für die Bedingung Level 88 in der SEARCH ALL-Anweisung hinzugefügt.
  - Unterstützung für REPORT SECTION wurde hinzugefügt.
- SQL

- Unterstützung für die DDL-Transformation aus der IBM IMS-Datenbank hinzugefügt.

AS400

Verbesserungen

- CL
  - Verbesserte Unterstützung von CL, indem die Transformation auf JAVA als Standardunterstützung festgelegt wurde
  - Die Umwandlung in Groovy-Skript wurde verbessert, indem die Verwendung eines Metamodells hinzugefügt wurde
  - Die Schlüsselwörter CONTINUE und FOR wurden zur Liste der reservierten Schlüsselwörter hinzugefügt
- DDS
  - Die Unterstützung von DSPF-Feldern, die im Vergleich zum vorherigen definiert wurden, wurde verbessert
  - Verbesserte Unterstützung für DDS-Zeichenfelder mit dem Schlüsselwort VARLEN
  - Verbesserte Transformation/Generierung von LF-Entitäten DAOs durch Generierung der expliziten Einstellung umbenannter Felder in den Vo-Methoden convert und update. JPAFrom
- RPG
  - Verbesserte Unterstützung des Schlüsselwortparameters PREFIX zur Behandlung von Fällen, in denen Werte keine Anführungszeichen enthalten
  - Verbesserte Behandlung von verschachtelten Methodenaufrufen in Array-Indizes
- COBOL400
  - Die Generierung von Datensatzentitäten, die mit der Anweisung COPY aus einem DSPF importiert wurden, wurde verbessert DDS-ALL-FORMATS
- SQL
  - Die Transformation von SQL-WHERE-Klauseln mit nicht eingeschlossenen OR-Bedingungen wurde verbessert.

## Versionshinweise 4.7.0

Veröffentlichungsdatum: 12. März 2025

Diese Version von AWS Blu Age Runtime and Transformation Engines enthält wichtige Aktualisierungen der wichtigsten Abhängigkeiten: AWS Blu Age Runtime wurde auf die neueste Spring Boot-Version 3.4.2 aktualisiert, und modernisierte Frontend-Anwendungen verwenden jetzt die neueste Angular Version 19.

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet. Andere Versionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 22.11.0 |
| | Npm 10.9.0 |
| | Eckig 19.1.3 |
| Serviceschicht | Spring Boot 3.4.2 |
| | Spring Core 6.2.2 |
| | Frühjahrssitzung 3.4.1 |
| Persistenzschicht | Spring Statemachine 4.0.0 |
| | PostgreSQL-Engine 14 |
| Oracle 21c | |
| Anwendungsserver | Apache Tomcat 10.1.17 |

## Runtime-Version 4.7.0

zOS

Neue Features

- Es wurde eine neue YML-Eigenschaft hinzugefügt, `dataSimplifier.doubleFloatingType` um zu definieren, welches Format beim Kodieren und Dekodieren der Gleitkommazahlen

verwendet werden soll. Zulässige Werte sind `IEEE_754` (Standard)/`HFP`(für das hexadezimale Gleitkommaformat)

- Unterstützung für die Dezimalskalierung der Position P auf dem numerischen BILD hinzugefügt

Verbesserungen

- COBOL
  - Verbesserte Unterstützung für die INSPECT-Anweisung mit den Klauseln AFTER INITIAL und BEFORE INITIAL
  - Verbesserte Unterstützung für numerisch bearbeitete Typen mit optimierter Bytekonvertierung zur Steigerung der Gesamteffizienz bei der Datenverarbeitung
  - Unterstützung für FUNCTION MIN & MAX mit alphanumerischem Vergleich hinzugefügt
  - Verbesserte Unterstützung für FUNCTION INTEGER für negative Werte
- PL/I
  - Verbesserte Unterstützung der PUT-Anweisung mit dem FBA/VBA SYSPRINT-Datensatz
- JCL - SORTIEREN
  - Unterstützung für die OUTREC-Option SQZ mit den Unterparametern SHIFT, LEAD, TRAIL, MID, PREBLANK und PAIR hinzugefügt
  - Verbesserte Katalogisierung von DCB-Attributen für die SORTOUT-Datei
- JCL - DSNUTILB
  - Verbesserte Unterstützung für den DECIMAL EXTERNAL-Datentyp
  - Verbesserte Unterstützung für die NULLIF-Bedingung, die sich auf die Werte/Position einer anderen Spalte bezieht
  - Verbesserte Unterstützung für BYTEA-Spalten
- JCL - INFUTILB
  - Verbesserte Unterstützung für Schlüsselwörter CURRENT [] DATE/TIME/TIMESTAMP im UNLOAD-Vorgang
- JCL — Verbesserte Unterstützung für den qualifizierten Rückgabecode von called PROCs
- Datenverarbeitung
  - Verbesserte numerische Vergleichslogik bei der Verarbeitung von Feldern mit leeren Werten, die von alphanumerischen zu numerischen Typen umdefiniert wurden, um eine korrekte Auswertung gegen Null zu gewährleisten

- Blusam
  - Verbesserte Unterstützung für den START-Befehl zur Bearbeitung von Teilschlüsselsuchen unter Verwendung von Segmenten des Primärschlüssels, wodurch flexiblere Funktionen zum Abrufen von Datensätzen bereitgestellt werden
- SQL
  - Unterstützung für Eingabeparameter, die in der ON-Klausel übergeben wurden, wurde hinzugefügt
  - Unterstützung für die REPEAT-Funktion hinzugefügt
  - Die Unterstützung für die UPDATE-Anweisung mit einer Zuweisung, die den Spaltennamen beinhaltet, wurde verbessert
  - Die Verarbeitung von SQLCODE und SQLSTATE bei mehrzeiligen Cursorabrufen wurde verbessert
  - Unterstützung für die DB2 XML-Funktion für Datenbankziele hinzugefügt DB2
- Integration der Nachrichtenwarteschlange
  - Verbesserte Fehlerbehandlung bei MQ GET-Vorgängen, wodurch eine klarere Statusanzeige MQCC_FAILED für leere Nachrichtentexte bereitgestellt wird
  - Verbesserte Unterstützung für MQ-PUT-Operationen durch Behandlung von MQRFH2 Headern im XA-Transaktionskontext
  - Verbesserte Unterstützung für AWS Blu Age Runtime Secrets zur Verwaltung zusätzlicher Eigenschaften für JMS MQ
  - Verbesserte Konfigurationskapazität für den JMS MQ-Verbindungspool
- Misc (Verschiedenes)
  - Systemintegration — Unterstützung für die Schemaumgebung in DFSRRC00-Programmaufrufen hinzugefügt
  - Kompatibilität für IMS-Transaktionen hinzugefügt, wenn Blusam deaktiviert ist
  - Die Unterstützung für die end-of-file Bedingung einer sequentiellen Datei wurde verbessert, um sie an die alte Regel „EOF wird als nicht erfolgreich gelesen“ anzupassen

AS400

Neue Features

- Unterstützung für den Befehl RUNSQLSTM zur Ausführung von SQL-Abfragen hinzugefügt, die in einer externen Quelldatei gespeichert sind

Verbesserungen

- CL
  - Es wurde ein Mechanismus zum Sperren von Dateien über die Befehle ALCOBJ/DLCOBJ hinzugefügt
  - Die CPYF-Befehlsfunktionalität wurde verbessert, um die Integrität der Datensatzsequenz beim Kopieren von Dateien aufrechtzuerhalten
  - Der CPYF-Befehl wurde verbessert, um die Ausgabe als Flatfile zu unterstützen
  - Verbesserte Parameterverwaltung zur automatischen Anpassung der Datenlänge bei Programmaufrufen
  - Verbesserte Parameterunterstützung für die Verarbeitung von Bereichsreferenzen
  - Unterstützung für MONMSG CPF21 05 im DLTDTAARA-Befehl hinzugefügt
  - Unterstützung der QTEMP-Bibliothek für den Befehl CHKOBJ auf DATAAREA hinzugefügt
  - Verbesserte Unterstützung für den Befehl ADDLFM, wodurch die Funktionen zur Verwaltung logischer Dateimitglieder verbessert werden
  - Unterstützung für generische Datenbereichsnamen im Befehl DLTDTAARA wurde hinzugefügt
  - Unterstützung für SBMJOB hinzugefügt, um Switches über den SWS-Parameter zu verarbeiten
  - Die Schreiblogik für den Datenbereich wurde verbessert, indem die Lese- und Schreibrufe in einem synchronisierten Block zusammengefasst wurden
  - Verbessertes Parallelitätsmanagement für den Datenbereich, um gleichzeitige Schreibvorgänge durch die Implementierung eines granularen Sperrmechanismus zu verhindern und Fehler bei parallel Vorgängen zu vermeiden
  - Unterstützung für das Schlüsselwort *ALL im DTAARA-Parameter für den Befehl RTVDTAARA wurde hinzugefügt
- Screen
  - Die Farbe eines weißen Attributs im Frontend wurde verbessert
- RPG
  - Verbesserte Unterstützung für den TESTB-Betrieb, um verschiedene Kodierungen besser handhaben zu können
  - Verbesserte Unterstützung für den EDITC-Betrieb mit den Bearbeitungscodes 'Y' und 'P'
  - Verbesserte Unterstützung für EDITCODE für die Codes K, Q und Z
  - Implementieren Sie die Unterstützung für die Unterstützung von ExtProc Prototype-Schlüsselwörtern in lokalen Prozeduren mit Parametern

  - Behandelte Parameter, die als Wert für externe Prozeduren übergeben wurden
- COBOL
  - Die SORT-Anweisung wurde verbessert, um die Zielbibliothek zu handhaben
  - Eine Partition wurde gekürzt, wenn das zugehörige DAO im OUTPUT-Modus geöffnet wurde
  - Unterstützung für das Schlüsselwort PREFIX wurde hinzugefügt
- DataQueue
  - Es wurden Keystore- und Truststore-Konfigurationsoptionen für RabbitMQ-Daten und hybride Nachrichtenwarteschlangen hinzugefügt, um sicheres Messaging in Produktionsbereitstellungen zu ermöglichen
  - Optimierte Leistung beim Serverneustart durch Eliminierung der redundanten Prüfung der Existenz von Datenwarteschlangen
  - Optimierte Architektur durch Entfernen von DataQueueRegistry Komponenten DataQueueConnectionProvider
  - Vereinfachtes Löschen von Datenwarteschlangen durch direkte Bearbeitung ohne Überprüfung der Registrierung
  - Verbesserter Prozess zur Erstellung von Warteschlangen durch Sicherstellung der ordnungsgemäßen Erstellung und queue-to-exchange Bindung von Exchange
- Misc (Verschiedenes)
  - Es wurde eine Überprüfung der Größe von Transaktionsanfragen hinzugefügt, um eine Systemüberlastung aufgrund übermäßiger Bereichsanforderungen zu verhindern
  - Verbessertes Sperrmanagement bei der Bereinigung von Arbeitsressourcen
  - Der DAO-Sperrmechanismus wurde verbessert, um zu verhindern, dass ein Datensatz gesperrt wird, wenn die Datei im INPUT-Modus geöffnet wird
  - Optimierte Leistung für die dynamische Zuordnung von Entitäten
  - Verbesserte Handhabung des JPA-Persistenzkontextes zur Verbesserung der Leistung bei DAO-Vorgängen
  - Verbesserte Datenbereichsunterstützung durch Hinzufügen von Zeitstempelspalten zur Verarbeitung von Datensatzmetadaten
  - Verbesserte Zuverlässigkeit des Datenbereichs durch Implementierung synchronisierter Lese- und Schreibvorgänge zur Vermeidung gleichzeitiger Zugriffsprobleme
  - Verbesserte Unterstützung für DateHelper #moveDate zur Unterstützung von Konvertierungen date-to-timestamp

- DDS
  - Verbesserte Unterstützung von Select/Omit-Bedingungen in Kombination mit synthetischen Operationen wie SST oder CONCAT
- SQL
  - Der SQLExecutor Builder wurde so optimiert, dass er das teilweise Abrufen von Datensätzen effizienter handhaben kann, insbesondere beim Abrufen mehrerer Zeilen mit einem Abruflimit von mehr als eins

## AWS Blu Age Transformationsengine 4.7.0

zOS

Verbesserungen

- COBOL
  - Unterstützung für die CODE-SET-Anweisung hinzugefügt
  - Unterstützung für die Option NOT END OF PAGE aus der WRITE-Anweisung hinzugefügt
  - Unterstützung für eine neue Syntax auf ALTER wurde hinzugefügt
  - Unterstützung für die Ersetzung mehrerer Zeilen für die Anweisung COPY REPLACING hinzugefügt
  - Verbessertes Parsen des CHARACTERS-Schlüsselworts, das in einer anderen Klauseldefinition impliziert wurde
  - Verbesserte Unterstützung für alternative Schlüssel, die im Eingabe-Ausgabe-Abschnitt verwendet werden, wenn die Namen der Schlüssel doppelt vorkommen, aber unterschiedliche übergeordnete Schlüssel haben
  - Verbesserte Unterstützung für IF-Bedingungen beim Vergleich großer Felder mit Leerzeichen
  - Die statische Auflösung von logischen Dateiverweisen wurde verbessert, da nun nicht mehr zwischen Groß- und Kleinschreibung unterschieden wird
  - Verbesserte Codetransformation zur automatischen Generierung impliziter LINAGE-COUNTER-Felder für jeden FD-Eintrag, der eine LINAGE-Klausel enthält
  - Verbesserte Unterstützung für die IDCAMS-Clusterdefinition mit einem Namen, der in einfache Anführungszeichen eingeschlossen ist
  - Verbesserte Unterstützung der PERFORM-Anweisung für das Basic-Format mit TIMES-Phrasenformat und Format, das die Qualifizierung von Abschnitten beinhaltet

  - Die Transformation der SORT-Anweisung mit mehreren GIVING-Dateien wurde aktiviert und eine Ausnahme hinzugefügt, um diesen Fall bei der Ausführung nachzuverfolgen
  - Die Transformation von RENAMES wurde verbessert, wenn sie für eine Gruppe mithilfe der REDEFINES-Klausel angegeben wurden
- PL/1
  - Support für die Option DATA oder LIST für PUT STRING hinzugefügt
  - Unterstützung für Programme mit mehreren Einträgen wurde hinzugefügt

AS400

Verbesserungen

- CL
  - Das Parsen von Parametern für QCMDEXC wurde verbessert, sodass ALCOBJ/DLCOBJ über QCMDEXC ausgeführt werden kann
  - Unterstützung ElseIf für Statements auf CL hinzugefügt
  - Unterstützung für die eingebaute Funktion %SWITCH zum Setzen der Job-Switches hinzugefügt.
- DDS
  - Die Unterstützung des im Programm beschriebenen DSPF-Datensatzes zur Verarbeitung von Dummy-Datensätzen in Eingabespezifikationen wurde verbessert
  - Die Bedingungen für die WHERE-Klausel der DAO-Generierung im Fall einer CREATE INDEX-SQLDDS-Datei wurden verbessert
- RPG
  - Unterstützung für die ZEROES-Initialisierung INZ (*ZEROS) auf Unsigned Integer hinzugefügt
  - Verbesserte Unterstützung des Befehls EXCEPT mit den Dateitypen Workstation und SQL Index
  - Verbesserte Unterstützung des internen RPG-Datentyps für extern definierte numerische Felder
  - Verbesserte Unterstützung des integrierten %ERROR zum Abrufen des aktuellen Fehlerstatus nach einer Operation an einer Datei
  - Verbesserte Unterstützung des Schlüsselworts COMMIT für die Verarbeitung logischer Dateien in mehreren Formaten
  - Verbesserte SELECT/OMIT-Verarbeitung für logische Dateien mit mehreren Formaten
  - Verbesserter Umgang mit MOVEA für das Setzen von Arrays mit Leerzeichen und das Verschieben fester Arrays von Packed, Zoned und Binary in ein anderes Array desselben Typs.

  - Verbesserte Array-Behandlung für Z-ADD *ZEROS und MOVEA *ZEROS
  - Verbesserte MOVE/MOVEL-Operationen beim Wechseln zwischen Zahlen- und Zeichenkombinationen
  - Implementieren Sie die Unterstützung für ExtProc RPGLE-Prototyp-Keywords in einer lokalen Prozedur mit Parametern
  - Die Handhabung der figurativen Konstante*HIVAL in Zuweisungsanweisungen wurde verbessert
  - Die Unterstützung für das Schlüsselwort PREFIX zur Zeichenersetzung wurde verbessert
- COBOL400
  - Verbesserte Unterstützung für REWRITE- und UPDATE-Operationen im dynamischen Zugriffsmodus
  - Unterstützung für die ACCEPT Statement FOR-Klausel in CBLLE wurde hinzugefügt
  - Die Unterstützung des integrierten %STATUS zum Abrufen des aktuellen Status einer Datei nach einer Operation wurde verbessert
- Misc (Verschiedenes)
  - Verbesserte Transformation von Programmen, die Datenstrukturen definieren, die denselben Namen wie eines der Felder im angegebenen PF/LF haben
  - Die Generierung von Array-Zugriffen auf Kanji-Variablen wurde verbessert

## Versionshinweise 4.6.0

Veröffentlichungsdatum: 24. Januar 2025

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet. Andere Versionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 22.11.0<br>Npm 10.9.0<br>Eckig 18 |
| Serviceschicht | Spring Boot 3.3.5 |

| | |
|---|---|
| | Spring Core 6.1.14 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 14 |
| | Oracle 21c |
| Anwendungsserver | Apache Tomcat 10.1.17 |

## Runtime-Version 4.6.0

zOS

Verbesserungen

- COBOL
  - Verbesserte `WRITE ADVANCING` Funktionen mit verbesserter Genauigkeit beim sequentiellen Schreiben von Dateizeilen, Unterstützung mehrerer Kontexte (`BEFORE>AFTER`, und implizite Verwendungen) und vollständige `PAGE` Implementierung von Anweisungen
  - Verbesserte Unterstützung `FILLER` für Fälle, in denen eine verschachtelte Tabelle FILLER als Gruppe mit einer untergeordneten Tabelle verwendet wird
  - Verbesserter Zugriff auf Kinder mit mehrdeutigen Eltern innerhalb eines Segments
  - Unterstützung für den Typ Numeric Edited mit picture='-----' wurde hinzugefügt
  - Die Anzeigebehandlung von Daten vom Typ BINARY wurde verbessert
- PL/I
  - Verbesserte Konvertierung von binären Literalwerten in Zuweisungsanweisungen
- JCL — SORTIEREN
  - Die Unterstützung für aufeinanderfolgende `OVERLAY` Parameter in derselben `OUTFIL` Anweisung wurde verbessert
- JCL — DSNUTILB
  - Optimierte Lademechanismen, was zu 25% schnelleren Datenabrufszeiten führt
  - Verbesserte Unterstützung für XA-Transaktionen für externe Geschäftsdatenquellen
- JCL — INFUTILB
  - UNLOAD — Unterstützung für den Datentyp hinzugefügt FLOAT8

- JCL — IDCAMS
  - Optimierter Umgang mit Rückgabecodes für Befehle `IDCAMS`
  - Unterstützung für das Löschen aller GDG-Generationen basierend auf dem GDG-Basisnamen hinzugefügt
  - Unterstützung für das Löschen von Dateien ohne Parameter hinzugefügt `NONVSAM`
- JCL — Verschiedenes
  - Verbesserte Verarbeitung von Batch-Neustart-Metadaten zur Verbesserung der Workflow-Statusverwaltung im Neustartmodus
- Blusam
  - Unterstützung von TTL für den Blusam-Cache wurde sowohl in Ehcache- als auch in Redis-Implementierungen hinzugefügt
  - Verbesserte Unterstützung für `DEPENDING ON` Felder in der COBOL-Datei Description für Blusam KSDS-Dateien `FD`
  - Verbesserte Thread-Sicherheit bei Redis Blusam-Lesevorgängen für die gleichzeitige Ausführung mehrerer Jobs
  - Verbesserte Erstellung von Blusam-Schemas für mehr Stabilität in Bezug auf Datenbankbenutzerrechte
  - Die Auffüllung nach rechts im verketteten Eingabedatensatz mit variablen Blöcken wurde verbessert `READ`
- ZURÜCK
  - Unterstützung für die Erstellung von Datensätzen im Multi-Schema-Modus hinzugefügt, einschließlich einer neuen Spalte „Schema“, die die Schemazuordnungen für jeden Datensatz angibt
- MFS
  - Verbesserte Übertragung von Benutzerinformationen vom Frontend zum gemeinsamen Kontext, wodurch eine korrekte Weitergabe an den JHDB-Kontext sichergestellt wird
  - Unterstützung für IBM MQ IMS-Informationsheader für XA-Transaktionen hinzugefügt
- SQL
  - Die `SQLCODE` Behandlung wurde verbessert, sodass beim Cursorabruf 305 gesetzt wird, wenn alle Spaltenwerte NULL sind
  - Unterstützung für `IN` Klauseln hinzugefügt, die `OCCURS` Parameter für `WHERE` Bedingungen enthalten

  - Unterstützung für `DECLARE GLOBAL` temporäre Tabellenanweisungen hinzugefügt
  - Erweiterte DB2 SQL-Unterstützung für das Zeitstempelformat, das für Mitternacht 24H DB2 spezifisch ist, und zwar durch spezielle Konvertierungen bei der Ausführung entsprechend der Zieldatenbank-Engine
- Misc (Verschiedenes)
  - Der Zeichensatz IBM93 0 wurde verbessert, sodass die beiden Unicode-Zeichen U+2014 und U +2015 X'44x4A' in EBCDIC entsprechen können
  - TDQUEUE — Die SQS-Implementierung wurde überarbeitet, um Multithreading zu unterstützen
  - Die Auflösung von GDG-Datensatznamen wurde verbessert, sodass der Kunde Dateien mit demselben GDG-Präfix archivieren kann (z. B. `A.B.C.G0002V00` ist es eine aktuelle Datei und ist eine Archivdatei) `A.B.C.G0001V00.1236`
  - Verbessert`SQLConverter::toPgmDate/Time/Timestamp`, um die Datumsberechnung an das alte Format anzupassen

AS400

Neue Features

- Unterstützung für dynamisch erstellte AS4 00-Tabellen für Flatfiles und duplizierte Entitäten hinzugefügt, wodurch der Zugriff auf Tabellen ermöglicht wird, die mit CL-Befehlen wie CRTPF, CRTDUPOBJ und CPYF erstellt wurden
- Es wurde ein Dienst hinzugefügt, der die Bibliotheksliste über eine Registrierung unterstützt, die die Standardbibliothek für jede Tabelle verwaltet

Verbesserungen

- CL
  - CLRPFM — Verbesserte Handhabung des Elements, wenn der Befehl für die QTEMP-Bibliothek aufgerufen wird
  - SMBJOB — Verbesserte Unterstützung von Parametern zur Verarbeitung dynamisch erstellter Argumente `PARM`
  - CPYFRMIMPF - Unterstützung für Parameter, und hinzugefügt `TIMFMT ERRRCDFILE ERRRCDOPT`

  - CPYFRMIMPF — Verbesserte Unterstützung für alphanumerische Datenbankwerte, die einfache Anführungszeichen enthalten
  - CPYF - Die Konstruktion von Befehlsabfragen für Dateien mit mehreren Mitgliedern wurde verfeinert `FROM TOMBR(*ALL)`
  - CPYF — Verbesserte Unterstützung für die Verarbeitung von Parametern `FMTOPT MAP DROP`
  - CPYTOIMPF - Verbesserte Unterstützung für die Bearbeitung von Parametern in der Tabelle `FROMFILE MEMBER`
  - RTVUSRPRF - Unterstützung für Parameter hinzugefügt `RTNUSRPRF`
  - DSPDBR — Der Befehl wurde überarbeitet, sodass er dem erwarteten Verhalten beim Ausdrucken von Informationen über Ansichten, die in einer Tabelle vorhanden sind, sowie über die Bibliothek und das Mitglied, zu dem sie gehören, entspricht
  - DSPFD — Verbesserte Unterstützung von Parametern `FILE`
  - DSPFD — Verbesserte Unterstützung der `TYPE MBR` Parameterausgabe um zusätzliche Werte: mbfile, mblib, mbfcdt, mfccn
- Screen
  - Verbesserte Priorität der Cursorposition für `DSPATR(PC)`
  - Die Validierung von Datensatzfeldern in Unterdateien wurde verbessert, indem die Frontend-Validierung von „geschützten“ Feldern ignoriert wurde
  - Verbesserte Unterstützung für die Initialisierung von Datensätzen in Workstations mit mehreren Array-Feldern, die Komponentennamen gemeinsam nutzen
  - Verbesserte Unterstützung für Antwortindikatoren in `DSPF` Schlüsselwörtern (`SFLMSGSFLMSGID`, `CHANGE` und Befehlstasten)
- RPG
  - Verbesserte Unterstützung für den Programmzyklus für eine bessere Handhabung der aus Primär- und Sekundärdateien gelesenen Felder
  - Unterstützung für Split Control Field für das Lesen von primären/sekundären Dateien hinzugefügt
  - Verbesserte `%SUBST` integrierte Methode zur Verarbeitung von Doppelbyte-Feldern in Vergleichsanweisungen
  - Verbesserte Unterstützung des ZERO-Indikators für den MVR-Betrieb
- DDS
  - Unterstützung für logische Dateien in mehreren Formaten im Datensatzformat, die sich auf denselben physischen Datensatz beziehen, wurde hinzugefügt

- DataQueue
  - Die Behandlung von Auftragsunterbrechungen für Jobs, die auf Datenwarteschlangennachrichten warten, wurde verbessert, indem der Benutzer bei Unterbrechungen bereinigt wurde
  - Für eine bessere Kanalverwaltung und Thread-Skalierung wurde von RabbitMQ zu Spring-AMQP migriert
- Misc (Verschiedenes)
  - Verbesserter SQLExecutor Builder zur Unterstützung von Abfragen mit mehreren Leerzeichen und offenen geschweiften Klammern ohne führende Leerzeichen
  - Verbesserte DAO-Unterstützung zur korrekten Handhabung der Cursorpositionierung bei Änderung der Leserichtung
  - Die Schlüsselinitialisierung nach Abruf- und Löschvorgängen wurde verfeinert, um sicherzustellen, dass zugehörige Datensätze vor dem Einfügen aktualisierter Datensätze ordnungsgemäß entfernt werden
  - Optimierter, von DAO Mapper generierter Code zur Verbesserung der Leistung bei der Zeitausführung

## AWS Blu Age Transformationsengine 4.6.0

zOS

Verbesserungen

- COBOL
  - Verbessertes Parsen von `RESERVE` Klauseln mit optionalem Literal `AREA/AREAS`
  - Verbesserte COBOL-Unterstützung mit optionaler `DATA DIVISION` Deklaration zur Unterstützung optimierter Testfälle
  - Der Absatz mit Sondernamen wurde verbessert, indem Unterstützung für`ALPHABET`, und `CLASS` -Klauseln`SYMBOLIC`, Schalter und Variablen hinzugefügt wurde `FORMFEED`
  - Unterstützung für `SYSIN` als mnemonischer Name in Anweisungen hinzugefügt `ACCEPT`
  - Verbesserte Unterstützung von `PICTURE` Klauseln für die Symbole „$“, „0", „CR“, „DB“ bei logischen Größenberechnungen `PIC`
  - Verbesserte `USE` Anweisungstransformation für Szenarien mit mehreren Dateien
  - Verbesserte `ALTER` Transformation von Anweisungen für mehrere Änderungen

  - Unterstützung für figurative Konstanten `ZERO` `HIGH-VALUE` `LOW-VALUES` in der Klausel hinzugefügt `delimited by`
- SQL
  - Die Transformation des Standardwerts für das PostgreSQL-Ziel wurde verbessert, sodass der Standardwert in Anführungszeichen umgewandelt wird `CURRENT_TIMESTAMP`
  - `WITH CHECK OPTION`Handle-Klausel von SQL-Ansichten

AS400

Verbesserungen

- DDS
  - Verbesserte Unterstützung von logischen Dateien in mehreren Formaten, die mehrfach auf denselben physischen Datensatz verweisen
- RPG
  - Verbesserte `MOVEL` Funktionen `MOVE` zur besseren Handhabung von Füll-Nullen
  - Verbesserter Umgang mit verschachtelten Funktionsaufrufen in Evaluationen und Bedingungen
- COBOL400
  - Unterstützung für die Umwandlung des `IN` Schlüsselworts in `SELECT` Anweisungen hinzugefügt
  - Die Unterstützung für fehlende Punkte in Datenbeschreibungseinträgen wurde verbessert und entspricht der neuesten COBOL-Version, bei der Punkte angenommen werden, wenn sie fehlen
  - Verbesserte Cursorpositionierung bei Operationen `REWRITE`
  - Verbesserte Unterstützung für Anweisungen `START` zum Sperren des Datensatzes an der aktuellen Dateiposition
  - Verbesserte Unterstützung für Compiler-Direktiven `COPY DDS` zur Generierung der gesamten Eingabe-/Ausgabedatenstruktur
- Misc (Verschiedenes)
  - StateMachines - Verbesserte Transformation zur Verbesserung der Deklaration zusammengesetzter Staaten im Einklang mit dem Stateless4j-Paradigma
  - Verbesserte Bereinigung für LF-Dateien, die Sonderzeichen enthalten
  - Verbesserte Unterstützung von figurativen `*ALL` mit Hexadezimalwerten
  - Verbesserte `MOVE` Operationsunterstützung für die implizite Konvertierung von numerischen in Zeichentypen

- Die Berichts-Bean-Generierung wurde optimiert, sodass sie nach dem zugehörigen Druckernamen sortiert wird, wodurch doppelte oder widersprüchliche Namen vermieden werden
- Verbesserte Unterstützung von Schlüsselwörtern in `EXTFILE` Kombination mit `USROPN` der Verwendung von Literalwerten und -formaten `libname/filename`

## Versionshinweise 4.5.0

Veröffentlichungsdatum: 20. Dezember 2024

Diese Version von AWS Blu Age Runtime und AWS Blu Age Transformation Engines umfasst die folgenden Hauptfunktionen.

- JCL-Unterstützung — Es ist jetzt möglich, JCL-Skripte im laufenden Betrieb innerhalb des Laufzeitkontextes zu generieren und auszuführen. Diese Funktion sorgt für mehr Flexibilität und Automatisierung bei der Batch-Auftragsverarbeitung. Wir haben die Unterstützung für JCL-Dienstprogramme in der Runtime aktualisiert und eine Reihe von Verbesserungen für SORT, ICETOOL, INFUTILB und IDCAMS eingeführt (Einzelheiten finden Sie in den folgenden Abschnitten). Diese Verbesserungen bieten robustere und effizientere Datenverarbeitungsfunktionen.
- Support von Bindungsverzeichnissen und Aktivierungsgruppen für modernisierte AS/400-Anwendungen — Bindungsverzeichnisse verbessern die Systemorganisation, indem sie exportierte Prozedurreferenzen verwalten, während Aktivierungsgruppen die Verwaltung des Ausführungskontextes optimieren. Diese Funktionen sorgen für mehr Präzision und Zuverlässigkeit, ein robustes Ressourcenmanagement und optimierte Systeminteraktionen. Das Ergebnis ist ein robusteres, organisierteres und effizienteres System für modernisierte AS4 00-Anwendungen.
- Aktualisierungen der Abhängigkeiten: — Aktualisierung aller Frontend-Frameworks (BAC/JAC und modernisierte Anwendungen) auf die Versionen mit langfristigem Support (LTS). Das Update von Angular von v17 auf v18 führt ein neues Reaktivitätsmodell und eine optimierte Statusverwaltung ein, wodurch die Komplexität reduziert und die Anwendungswartung für Entwickler verbessert wird. Node.JS wurde ebenfalls von v20 auf v22 aktualisiert.

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet. Andere Versionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |

| | |
|---|---|
| Java | Java 17 |
| Präsentationsebene | Knoten JS 22.11.0 |
| | Npm 10.9.0 |
| | Eckig 18 |
| Serviceschicht | Spring Boot 3.3.5 |
| | Spring Core 6.1.14 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 14 |
| | Oracle 21c |
| Anwendungsserver | Apache Tomcat 10.1.17 |

## Runtime-Version 4.5.0

zOS

Neue Features

- JCL — Es wurde die Möglichkeit hinzugefügt, einen Batch-Job aus Online-Programmen aufzurufen. Wir haben einen Dienst für die Verarbeitung von JCL-Skripten hinzugefügt, die in einem speziellen Programm gespeichert sind, TDQueue wenn sie von einem modernisierten Programm im laufenden Betrieb generiert werden. Dieser Dienst ermöglicht die Rekonstruktion der JCL-Meldung, wandelt diese Nachricht in ein Groovy-Skript um und führt dieses Groovy-Skript aus.
- ADABAS — Unterstützung für das ADABAS-Programm hinzugefügt. Mit dieser Unterstützung emuliert die Runtime die ADABAS-Befehle für den Datenbankzugriff (nur für Oracle verfügbar).

Verbesserungen

- COBOL
  - Verbesserte Unterstützung der DISPLAY-Anweisung unter Nutzung der Option NO ADVANCING

  - Höhere Genauigkeit bei der Verwaltung von Währungszeichen, sodass der Benutzer von einer genaueren transformierten COBOL-Struktur profitieren kann
  - Verbesserte Unterstützung für die Wertzuweisung beim Verschieben eines Felds ohne Vorzeichen in ein Feld mit Vorzeichen und umgekehrt
  - Verbesserte Unterstützung der Blockgröße für GDG-Dateien und verkettete Dateien
- CICS
  - Unterstützung für OpenStatus und EnableStatus von Blusam-Datensätzen hinzugefügt
  - Unterstützung für den Befehl wurde hinzugefügt `SET DATASET`
- JCL — SORTIEREN
  - Verbesserter Umgang mit der Datensatzgröße
  - Verbesserte Unterstützung für die `OUTFIL` Anweisung, Ausgabedateien zu erzeugen, die nur die Datensätze aus den Eingabedateien gemäß den in STARTREC und `ENDREC` Optionen angegebenen Werten enthalten
  - Verbesserte Unterstützung von Anweisungen `OVERLAY`
  - Die Unterstützung für die `OUTREC` Anweisung zur Behandlung einer Variante der `EDIT` Option wurde verbessert. Wir unterstützen jetzt `EDIT(...)` zusätzlich `EDIT=(...)`
  - Unterstützung für das Muster (p, m, f, OPERATOR, p2, m2, f2) in arithmetischen Operationen hinzugefügt
  - Sie können die `DUMMY` File-Klausel des `SORT` Programms aus einer JCL verwenden, um leere Eingabedateien zu verarbeiten und von der Generierung leerer Dateien zu profitieren
- JCL — ICETOOL
  - Verbesserte Unterstützung für das `SORT FIELDS=COPY` Statement durch das Programm `SORT`
- JCL — INFUTILB
  - Verbesserte Unterstützung für die Berechnung von Datensatzgrößen, wenn sie nicht in der JCL angegeben ist und die Eigenschaft DFSIGDCB deaktiviert ist
  - Die UNLOAD with INTO-Klausel für DECIMAL wurde verbessert, indem Genauigkeit und Skalierung entsprechend den Feldern der INTO-Klausel aktualisiert wurden
  - Verbesserte Formatierungsmethode in VarcharFormatter
  - Verbesserte Unterstützung mit einer neuen konfigurierbaren Option, mit der Benutzer steuern können, wie VARCHAR-Felder beim Entladen von Daten behandelt werden, was das Füllverhalten angeht, wodurch Flexibilität und Genauigkeit bei Datenextraktionsprozessen gewährleistet werden.

- JCL — IDCAMS
  - Das Löschen von Dateien mit Platzhalter-Suffix und Namen, die entweder direkt in Klammern oder in einfachen Anführungszeichen definiert sind, wurde verbessert
  - Verbesserte Genauigkeit bei der Nutzung des MAXCC-Rückgabecodes
- JCL — IKJEFT01 — Feature-Flag `systin.encoding` (default =`ASCII`) hinzugefügt, um spezifische Kodierungen für SYSTSIN-Dateidatensätze zu unterstützen
- JCL — Verbesserte Unterstützung der BDW-Eigenschaft für eine generierte Ausgabedatei in einem JCL-Schritt und nachfolgende Schritte verwenden dasselbe Dateisystem als Eingabe und `DISP=PASS`
- MF
  - Verbesserte Unterstützung für 2-Byte-Header für Record Sequential-Dateien
  - Die Behandlung von Rückgabecodes für den Befehl DELETE wurde verbessert
  - Die Zeile für den Schreibfortschritt bei sequentieller Datensatzdatei wurde verbessert
- Redis
  - Die Initialisierung von Redis-Vorlagen für JCL-Checkpoints und Jics wurde verbessert TSQueues
  - Verbesserte Zugänglichkeit und Lesbarkeit von Informationen zu Sperren von Redis-Datensätzen
- SQL
  - Verbessertes Parsen von FOREIGN KEY mit der REFERENCES-Klausel
  - Es wurde eine erweiterbare Caching-Funktion bereitgestellt, um ursprüngliche ältere Grafiktypen in der Datenbank zu speichern, wodurch die Rückverfolgbarkeit von Daten verbessert und die grafische Berechnung erleichtert wird
  - Verbesserte Parsing-Unterstützung für das CASE WHEN-Muster von SQL-Abfragen in allen Runtime-Dienstprogrammen
  - Verbesserte in SQL Postgres Blu Age integrierte Funktion gwdecimal, auf die sich die Laufzeit stützt, damit sie zur integrierten DECIMAL-Funktion passt. DB2
- Misc (Verschiedenes)
  - Verbesserte Unterstützung für die Verwendung des SIGN-Operanden NumericEditedType
  - Verbesserte Generierung der Konfiguration der primären Datenquelle SpringBootLauncher in der modernisierten Anwendung
  - Verbesserte Flexibilität bei der Trennung von Anwendungsprotokollen und Pfaden, die sich auf den aufgerufenen Job beziehen.

  - Verbesserte Unterstützung für einen leeren Wert beim Vergleich von Feldern NumberUtils
- DATEI — Verbesserte Unterstützung von Datensätzen mit variablen Blöcken in den zugrunde liegenden Dateien
- MQ — Verbessertes MQ-Verbindungsmanagement für Hochverfügbarkeitsumgebungen bereit
- Verbesserte MQ Queue-Kompatibilität durch zusätzliche Unterstützung für Nicht-JMS-Clients zur Verbesserung der Kodierung und Zeichensatzverarbeitung
- Verbesserte Unterstützung für ANSI-Steuerzeichen für Ebcdic-Dateien

AS400

Neue Features

- Unterstützung für exportierte Daten innerhalb gebundener Programme hinzugefügt
- ILE-spezifische Unterstützung für die Division durch Null hinzugefügt

Verbesserungen

- COBOL400
  - Verbesserte Unterstützung von EOF im Dateistatus
  - Erhöhen Sie die Präzisionsunterstützung der Cobol START-Anweisung, um das Schlüsselwort EQUAL in der KEY IS-Klausel zu unterstützen
- CL
  - Unterstützung für den Befehl UPDENVPARM wurde hinzugefügt
  - CRTPF — Unterstützung für Tabellen hinzugefügt, auf die über eine Partition zugegriffen wird
  - RCVF — Verbesserte Unterstützung für logische Dateien mit Override
  - FTP — Verbesserte Unterstützung logischer I/O-Dateien mit OVRDBF und erweitertem OUTPUT-Protokoll sowie zusätzliche Unterstützung für I/O-Dateien im Arbeitsverzeichnis
  - CPYFRMIMPF - Unterstützung für Parameter hinzugefügt, `ERRRCDFILE` `TIMFMT` `ERRRCDOPT`
  - CPYF - Verbesserte Erstellung von QTEMP-Partitionen
  - CPYF - Es wurde eine Überwachungsmeldung hinzugefügt, wenn die *FROM-Datei leer ist
  - OVRPRTF - Unterstützung für neue Parameter hinzugefügt:`PAGESIZE`,,,,,`OUTQ`, `DEV` `LIP` `CPI` `OVRFLOW` `LVLCHK` `FORMTYPE` `HOLD`

  - Höhere Genauigkeit bei der Verwendung des FMTOPT Parameters mit MAP und DROP Optionen im CPYF Befehl, um das Kopieren von Daten aus einer Quelldatei mit zusätzlichen Spalten in eine Zieldatei zu ermöglichen
  - Höhere Genauigkeit bei der Verwaltung der Zuordnung von Platzhaltermustern für Dateisystempfade im Befehl RMVLNK
  - Der Befehl RMVM (Remove Virtual Machine) wurde dahingehend erweitert, dass er DROP Partitionstabellen verarbeitet und so eine vollständige Bereinigung der zugehörigen Ressourcen gewährleistet.
  - OPNQRYF — Verbesserte Unterstützung des Parameters *FILE für den Befehl
  - Die CPF0000-Behandlung wurde implementiert, um alle Nachrichten zu umfassen CPFx
  - CHGDTAARA — Unterstützung für das Schlüsselwort *ALL hinzugefügt, um den gesamten Datenbereich zu ändern
- Screen
  - Der tables/subfile displaying by increasing accuracy for scrolling and position/priority Cursor wurde verbessert
  - Verbesserte CHECK(RZ) CHECK(RB) Funktionalität für nicht numerische Felder und Felder ohne Vorzeichen
  - Verbesserte Unterstützung der Hilfescreen-Funktion für Stichwörter HLPARA
- RPG
  - Verbesserte Unterstützung von Built-In %SubDt
  - Verbesserte Unterstützung für Verfahren, die eine lokale Datenstruktur verwenden, die extern beschrieben wird
  - Unterstützung für optionale Fehlercode-ParameterQMHSNDPM,, und QMHRMVPM hinzugefügt QMHRCVPM
  - Verbesserte Unterstützung der %SUBST integrierten Methode zur besseren Verarbeitung von Doppelbyte-Feldern.
  - Unterstützung für das integrierte %TLOOKUP und seine Varianten (%TLOOKUPGE, %TLOOKUPGT, %TLOOKUPLE, %TLOOKUPLT) hinzugefügt
- Datenbereich
  - Verbesserte Unterstützung für den OUT-Betrieb, wenn Factor1 leer ist
  - Verbesserte gleichzeitige Lesevorgänge im gleichen Datenbereich

  - Es wurde eine Konfigurationsvariable hinzugefügt`blu4iv.dtaara.library.disable`, um Bibliotheken für den Datenbereich zu deaktivieren
  - Erweiterte Unterstützung für die Nutzung benannter Bibliotheken durch Operationen im Datenbereich, sodass der Benutzer die Position des Datenbereichs nach Belieben strukturieren kann.
- DataQueue
  - Verbesserte Nutzung des RabbitMQ-Kanals
  - RabbitMQ Consumer wurde verbessert, sodass nun nur noch einmal versucht wird, den Kunden zu kündigen
  - Der Abruf der Datenwarteschlange aus RabbitMQ wurde verbessert, indem BasicGet nur dann versucht wird, wenn die Wartezeit 0 ist
- Misc (Verschiedenes)
  - Benutzerbereich — Verbessertes Verhalten, wenn mehrere Jobs versuchen, denselben Userspace gleichzeitig abzurufen
  - Die Unterstützung für das Löschen von Datensätzen ohne Commitment Control wurde verbessert
  - Entität — Verbesserte Unterstützung für aufeinanderfolgende Auslassungen, da OMIT eine implizite Bedeutung hat `AND`
  - Unterstützung für Camelcase in Entitäten, Mapper und Setter hinzugefügt, um benutzerdefinierte Namen zu handhaben, die durch zusätzliches Refactoring definiert wurden
  - Verbesserte Weitergabe von Benutzerinformationen aus AS4 00 Umgebungstransaktionen durch die gesamte Anwendung.
  - Verbesserte Genauigkeit beim Beenden eines von Quartz geplanten Jobs im Falle einer Unterbrechung.
  - Die Unterstützung von Commitment Control wurde verbessert, um den Umfang des Programms zu verbessern

## AWS Blu Age Transformation Engine 4.5.0

zOS

Verbesserungen

- JCL — Verbesserte Groovy-Generierung für KSDS-Datensätze basierend auf LISTCAT-Parsing

- COBOL
  - Die Analyse der `COPY-REPLACING` Anweisung wurde verbessert, um das Ersetzen eines qualifizierten Unterfeldes zu handhaben, wenn für diesen Unterfeldnamen eine Mehrdeutigkeit besteht
  - Verbesserte Unterstützung für Defined in Statement `SYSOUT` `SPECIAL-NAMES`
  - Verbesserte Unterstützung von figurativen NULLEN in Statements `ADD n TO ZERO`
  - Verbesserte Unterstützung für Anweisungen zur `REPLACE` Behandlung mehrzeiliger Probleme durch das Reduzieren mehrzeiliger Tasten und Textblöcke
  - Verbesserte Unterstützung für arithmetische Operationen mit Klausel ADD/SUBTRACT/ MULTIPLY/DIVIDE `GIVING`
  - Die Parsing-Unterstützung von REPORT SECTION und den zugehörigen Aktionen (INITIATE, TERMINATE, GENERATE report) wurde eingeleitet
- Verschiedenes — Verbessern Sie die Generierung und Robustheit von Wetterberichten

AS400

Verbesserungen

- DDS
  - Verbesserte Unterstützung der impliziten Länge des Typs DATE
  - Verbesserte Unterstützung von stop-zero-suppression Zeichen im Schlüsselwort EDITWORD
  - Verbesserte Unterstützung des Spaltennamens DESC, da es sich um ein reserviertes Wort in DB handelt
- RPG
  - Verbesserte Unterstützung des integrierten %TIME
  - Verbesserte Generierung von EVALR-Anweisungen, um die Zuweisung von einem Zeichenkettenwert zu einer Variablen kürzerer Länge mit einer besseren Rechtsanpassung zu handhaben
  - Verbessertes SQL-Parsing rund um die Einstellung von Optionen
  - Verbesserte Unterstützung für die PSDS-Initialisierung in NOMAIN-RPGLE-Programmen
  - Verbesserte Unterstützung für das Schlüsselwort LIKE, um ein numerisches DDS-Feld unabhängig von seiner externen Beschreibung als Packed zu definieren
  - Die Bereinigung von Dateinamen wurde verbessert, indem „$“ durch „DL“ ersetzt wurde

    - Die Unterstützung des integrierten %SUBST zur Verarbeitung von Doppelbyte-Werten wurde verbessert
- COBOL400
    - Bildschirm — Verbesserte Unterstützung von DSPF-Datensätzen rund um I/O-Operationen
- CL
    - Die Umbenennung von reservierten Variablennamen wurde verbessert
    - Verbesserte Unterstützung von Select/Omits-Bedingungen zur Verarbeitung von Dateien mit mehreren Dateiformaten
- Misc (Verschiedenes)
    - Reduzierung doppelter Entitäten im Zusammenhang mit Dateioperationen (EOF, FOUND, EQUAL)
    - Verbesserte Generierung von JRXML-Dateien für QPRINT, einen Standarddrucker auf AS/400. Wenn sie verwendet wird, enthält die erzeugte JSON-Datei keinen Verweis auf das Programm oder die Datei. Es wird nur eine JRXML-Datei generiert (QPrint-QPrint.jrxml)
    - Die Anzeige zusätzlicher Nachrichteninformationen für Komponenten, die Nachrichten aus der Programmwarteschlange anzeigen, wurde verbessert

## Versionshinweise 4.4.0

Veröffentlichungsdatum: 13. November 2024

Diese Version von AWS Blu Age Runtime and Transformation Engines konzentriert sich auf die Aktualisierung kritischer Abhängigkeiten und unterstützter Technologien bei gleichzeitiger Steigerung der Leistung mehrerer Funktionen. Zu den wichtigsten Funktionen und Änderungen in dieser Version gehören:

- Aktualisierungen der Abhängigkeiten: Konsolenanwendungen (BAC und JAC) und modernisierte Anwendungen laufen jetzt auf Bootstrap 5. Die AWS Blu Age Runtime wird jetzt vom Spring Boot 3.3.5-Framework unterstützt.
- Leistung: Dank einer neuen Implementierung, die Leistungseinbußen nach dem Upgrade der Spring State Machine-Bibliothek von Version 2.5.1 auf 4.0.0 überwindet, wurde die Leistung der Ausführung von Zustandsmaschinen verbessert (bis zu 10x schneller). Dieses Upgrade war nicht optional, da die Version 2.5.1 nicht mehr gewartet wurde und Critical und High enthält. CVEs Es umfasst eine Runtime-State-Machine-Implementierung auf der Plattform bis hin zu einer neuen

Bibliothek mit einer schlanken und effizienten State-Machine-Implementierung, die frei von CVE ist und insgesamt eine bessere Leistung bietet.

- Vereinfachung des Datenbankzugriffs: Es wurde eine umfassende Überarbeitung der für den Datenbankzugriff verwendeten Komponenten abgeschlossen, darunter JPA-Entitäten DAOs, DDS-Entitäten und Mappers DataSimplifier . Diese Neugestaltung war auf die Notwendigkeit zurückzuführen, die in 00-Projekten übliche OVRDBF-Funktion (Override Database File) besser zu unterstützen. AS4 Es ermöglicht die Behandlung von mehr Fällen mit einer vereinfachten Architektur für den generierten Code.

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet. Andere Komponentenversionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 18.18 |
| | Npm 9.8 |
| | Eckig 17 |
| Serviceschicht | Spring Boot 3.3.5 |
| | Spring Core 6.1.14 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 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 4.4.0

zOS

Neue Features

- COBOL — Unterstützung für die JSON GENERATE-Anweisung hinzugefügt
- COBOL — Unterstützung für Steuerblöcke hinzugefügt
- MF — Unterstützung für die FCDREG-Compiler-Direktive hinzugefügt
- Blusam - Funktion VSAM-Dateisätze mit einer Implementierung hinzugefügt, die auf dem Datenbankschema basiert - Nur PostgresSQL wird unterstützt
- Blusam — Unterstützung für den Umgang mit TTL (Time to Live) für zwischengespeicherte Blusam-Datenelemente hinzugefügt (Redis-Cache-Engine)
- JCL - IDCAMS - Neue Eigenschaft hinzugefügt, um den für die Dekodierung der SYSIN-Karte verwendeten Zeichensatz zu erzwingen `idcams.encoding.forced`
- JICS — Die `jics.db.dataScriptLocation` Eigenschaft wurde von nun an so erweitert, dass sie eine Liste von `application-main.yml` Datei- und Ordnerpfaden akzeptiert. Die Reihenfolge der Liste ist wichtig. Die erste SQL-Datei wird zuerst ausgeführt und so weiter. Wenn ein Ordner ausgeführt wird, werden die darin enthaltenen SQL-Skripts in keiner bestimmten Reihenfolge ausgeführt.
- Unterstützung für das CEE3 ABD-Hilfsprogramm hinzugefügt

Verbesserungen

- Blusam - Verbesserte Ladezeit und Speicherbedarf von älteren großen Datensätzen auf Blusam für Kunden, die die PostgreSQL-Engine verwenden (wir haben eine bis zu 8-fache Erhöhung der Ladegeschwindigkeit für große Datensätze beobachtet)
- Blusam - Verbesserte exportDataSet ToS3-API mit Support für Anmeldeinformationen
- Blusam — Verbessertes LISTCAT-Hochladen von Dateien für die Erstellung von Datensätzen
- Blusam — Verbesserte Unterstützung für dynamisches Lesen mit explizitem KEY
- Blusam — Die Logik des Write-Behind Mechanismus wurde verbessert
- JCL - Verbesserte JES-Unterstützung zur Verbesserung der Dateisperre bei parallel Ausführung
- JCL — Unterstützung für Statements hinzugefügt `INCLUDE MEMBER`
- JCL - DNSUTILB - Verbesserte Unterstützung für doppelte Schlüssel zur Behandlung von Sonderfällen, wenn der Primärschlüssel Leerzeichen enthält

- JCL - DSNUTILB - Verbessert, um die Leistung beim Laden von GRAFIKdaten zu optimieren LoadTask
- JCL - INFUTILB - Unterstützung für „Wenn nicht definiert“ hinzugefügt `fetchsize` `chunksize`
- JCL - INFUTILB - Verbesserte Unterstützung für Abfragen, die einen leeren Ergebnissatz zurückgeben
- JCL - INFUTILB - Verbesserte Robustheit bei der Verarbeitung von Daten in CHUNK
- JCL - INFUTILB - Verbesserte Unterstützung für das Entladen mit einem Feld, das Nullwerte zulässt
- JCL - INFUTILB - Verbesserte Unterstützung für numerische Typen
- JCL - INFUTILB - Verbessertes Entladen für ein Feld, das Nullwerte zulässt
- JCL - SORT - Verbesserte Unterstützung für die OUTREC-Syntax
- JCL - SORT - Verbessertes Parsen der Anweisung DATE1
- JCL - SORT - Verbesserte Unterstützung der Klausel INREC PARSE mit RDW
- JCL - SORT - Verbesserte Formatierung von Feldern mithilfe von Bearbeitungsmasken
- JCL - SORT - Verbesserte Unterstützung von 'SubString' in OUTREC
- JCL - SORT - Verbesserte Unterstützung für MF-kompatible CARDS
- JCL - UNLOAD - Verbesserte Unterstützung der Feldgröße mit Postgresql
- JCL - IDCAMS - Verbesserte Leistung beim Laden von VSAM-Datensätzen durch Einführung des Bulk-Modus
- PL/1 — Verbessert die Unterstützung für NumericEditedType Formatierungen, um Skalierungsabweichungen zu vermeiden
- IMS — Verbesserte Unterstützung für IMS-Datenbank _rechte Spalte in NodeSorter
- CICS — Verbesserter Befehl `RECEIVE MAP` mit `SET` und ohne `INTO`
- BMS — Verbesserte Unterstützung des Feld-Anfangswerts
- SQL — Verbessertes DateTimeFormat Parsen nach Mustern `ddMMMyy`
- COBOL — Verbesserte Unterstützung für NumericEditedType Werte, bei denen das Dezimalkomma beim Abrufen des Werts nicht berücksichtigt wird
- Verbesserte Unterstützung für das Lesen von Feldern mit variabler Länge in einer zeilensequentiellen Datei
- Verbesserte Unterstützung für die Vererbung von Datensatzgrößen aus dem Datensatzkatalog für GDG-Dateien

- Die Unterstützung für das Drucken von Berichten wurde verbessert, da nun anpassbare Vorwärtslinien möglich sind
- Verbesserte Initialisierung von Datensatzdaten für VB-Dateien (Variable Block)

GS21

Neue Features

- Bildschirm — Unterstützung für PSAM-Dateien hinzugefügt
- Bildschirm — Unterstützung hinzugefügt für ATTR2
- Unterstützung für das AIM-Ökosystem (Advanced Information Manager) hinzugefügt.
- PED-Unterstützung in AIM hinzugefügt

Verbesserungen

- Verbesserte Handhabung von BitUtils Signaturen RangeReference
- Verbesserte Unterstützung für das Hinzufügen von DummyFileConfiguration Attributen recordSize/rdw/bdw/blksize/blkszlim
- Verbesserte Unterstützung für die VPOINT-Anweisung zur Behandlung des Falls ein Datensatz nicht gefunden wurde
- Robustheit beim Zugriff auf das Datensatz-Byte-Array wurde hinzugefügt
- Verbesserte JEF-Zeichensatz-Zeichenzuordnung
- Verbesserte Unterstützung für den Umgang mit Arrays und Bedingungen beim JDBC-Mapping
- Verbesserte Unterstützung für SQL-Anfragen in den verschiedenen NDB-Anweisungen, bessere Handhabung der Variationen von SQL-Syntaxen mithilfe von Konstanten für jeden Teil einer SQL-Abfrage.
- Verbesserte Unterstützung für die numerische GS21 PackedType Überprüfung, ob der letzte Nibble C, D oder F sein soll
- Bildschirm — Verbesserte Unterstützung für ACSAPI sowie DefaultPsamController für SPA und ENTER
- Bildschirm — Verbesserte Unterstützung von ACSAPI- und NDB-Verben

### AS400

#### Neue Features

- Unterstützung für Datenbankdateien im Format mit mehreren Datensätzen hinzugefügt
- Das AS4 00-Framework für den Datenbankzugriff wurde neu gestaltet
  - Verbesserte Funktionen rund um das Überschreiben von Dateien
  - Veraltete Komponenten wurden entfernt und die Komplexität reduziert
  - Der aus älteren Programmen generierte Code wurde optimiert
  - Die DAOCycle Manager-Komponente wurde in das Blu4IV-Plugin integriert, sodass wir die AS4 00-spezifischen Funktionen unserer benutzerdefinierten Runtime nutzen können.
- JOB — Verbesserte Unterstützung für das Job-Management (Quartz), um die Möglichkeit zu bieten, einen Job/eine Gruppe von Jobs zu unterbrechen. Es wurde ein REST-API-Endpunkt hinzugefügt, um einen Job mit der angegebenen Ausführungs-ID zu unterbrechen (für jeden Job eindeutig, da es sich um einen Primärschlüssel handelt). Bei erfolgreicher Unterbrechung aktualisiert die Runtime den Status des Jobs auf „INTERRUPTED“.
- Unterstützung für das Hilfsprogramm CEERAN0 wurde hinzugefügt
- Unterstützung für den passiven Modus hinzugefügt. YAML wurde hinzugefügt`configuration gapwalk-application.cl:ftpservice:passive`, um den passiven Modus zu aktivieren
- Funktion hinzugefügt, um QTEMP-Sitzungen zu erstellen und die QTEMP-Reinigung zu verzögern
- Unterstützung für die Kompilierungsfunktion BNDDIR hinzugefügt, um explizite Abhängigkeiten zwischen Programmen zu definieren
- Unterstützung für den Aktivierungsgruppen-Mechanismus wurde hinzugefügt

#### Verbesserungen

- CL — Der Befehl RMVMSG in der Nachrichtenwarteschlange des Programms wurde verbessert, um das Schlüsselwort *PREV zu verarbeiten
- CL — Verbesserte Unterstützung für Overrides in OPNQRYF
- CL — Unterstützung für die Parameter MSGLEN und SECLVLLEN für den Befehl RTVMSG hinzugefügt
- CL — Verbesserte Unterstützung für CRTDUPOBJ zur Verwaltung von Fällen, in denen NEWOBJ nicht übergeben wird, und Unterstützung für generische Tabellennamen hinzugefügt

- CL — Verbesserte FTP-Unterstützung für die Verarbeitung der Parameter GET, RMTSYS und BINARY
- CL — Die CLRPFM-Abfrageleistung wurde verbessert und es wurde eine Option hinzugefügt, um TRUNCATE anstelle von DELETE zu verwenden
- CL — SBMJOB wurde verbessert, um den USER-Parameter korrekt zu handhaben und ihn als USER zu verwenden, wenn ein Job eingereicht wird
- CL — Die Unterstützung von DLTOVR-Befehlen wurde verbessert, um den Fall *ALL zu behandeln
- Data Area — Verbesserte Unterstützung für Blu4 DataArea durch Hinzufügen von Protokollen für die Ausnahmebehandlung
- Datenbereich — Verbesserte Unterstützung für Blu4DataArea , um für jeden Thread eine neue DataAreaDao Instanz abzurufen
- Datenbereich — Verbesserte Datenbereichssperren, wodurch Sperren auf Datensatzebene vermieden und stattdessen der neu implementierte Sperrmechanismus verwendet wird
- Datenbereich — Der Schreibvorgang für den Datenbereich wird jetzt mit der Ausführung fortgesetzt, wenn keine Sperre aktiviert wurde und eine Fehleranzeige angezeigt wird
- Bericht — Verbesserte Unterstützung für Ausgabepfade und Benennungskonventionen für Berichte in gedruckten Berichten. Kunden konnten auch den Ausgabepfad und den Namen des Berichts anpassen. Der Kunde kann seinen eigenen Pfad und seine eigene Namenskonvention angeben, ohne dass dies Auswirkungen auf andere Projekte hat.
- JOB — Verbesserte Unterstützung für das Job-Management (Quartz) zur Aktualisierung des Jobstatus im Falle einer abnormalen Jobbeendigung, z. B.: „Shutdown“ oder „abnormal Shutdown“ von Tomcat
- Bildschirm — Verbesserte Handhabung von numerischen Werten in Feldern mit einem Bearbeitungswort mit Minuszeichen
- Bildschirm — Verbessertes Render-Popup mit nur titleColorTop
- Bildschirm — Verbesserte Unterstützung für das Abrufen von Hilfeinformationen für den Fall, dass ein allgemeines Hilfeelement nicht gefunden wurde
- Bildschirm — Die Anzeige des Bildschirms „Zusätzliche Informationen“ wurde verbessert, wenn in der Nachrichtenzeile der Unterdatei F1 gedrückt wird
- Bildschirm — Verbesserte Anzeige der Fußzeilen in Nachrichtenzeilen für SFLMSG
- Bildschirm — Das Frontend wurde verbessert, sodass ein Datensatz vollständig entfernt wird, wenn sich ein neuer Datensatz mit ihm überschneidet

- Warteschlangen — Der RabbitMQ-Nachrichtenabruf wurde verbessert, sodass weniger Ressourcen verbraucht werden
- Queuing — Die Implementierung von RabbitMQ Data Queue wurde verbessert, sodass jeweils nur eine Nachricht abgerufen wird.
- SQL — Verbesserte SQLExecutor Builder-Behandlung von SQLCODE für dynamische CREATE- und DROP-Tabellenabfragen
- SQL — Verbesserte Unterstützung von OVRDBF bei Abfragen
- SQL — SQLExecutor Der Builder wurde verbessert, sodass OVRDBF-Überschreibungen auf vorbereitete Anweisungen angewendet werden
- RPG — Verbesserte Unterstützung für Eingabe- und Ausgabespezifikationen von vom Programm beschriebenen Festplattendateien
- RPG — Verbesserte Unterstützung für das Lesen primärer und sekundärer Dateien mit dem MR-Indikator (Matching Records). Die Abrufreihenfolge eines DAO-Zyklus mit Abgleichsfeldern wurde verbessert.
- RPG — Verbesserte Unterstützung für Primär- und Sekundärdateien. Verbesserung des Updates: Primärdateien und Ausgabe. Generierung von Sekundärdateien zum Aktualisieren/Schreiben von Code.
- RPG - Unterstützung für RETURN-Anweisungen im Freiformformat hinzugefügt
- RPG — Verbesserte Transformation und Laufzeitbehandlung von numerischen Dezimalzuweisungen,
- RPG — Verbesserte Generierung von binären Variablen
- RPG — Verbesserte Unterstützung für EDITC
- RPG — Verbesserte Handhabung des lokalen Datenbereichs
- Verbesserte Unterstützung von DDS-Feldern, die von mehreren Gerätetypen (DISK, WORKSATION, PRINTER) gemeinsam genutzt werden
- Die Behandlung von Überschreibungen wurde verbessert, sodass aktivierte Überschreibungen keine Auswirkungen PFs mehr haben LFs
- Blu4 wurde verbessert, sodass der Benutzername und die Benutzer-ID ivWebController nicht auf die Standardwerte zurückgesetzt werden
- Verbesserte Indexanpassung beim Lesen von Datensätzen, wenn sich die Leserichtung ändert
- Verbesserte Cursorplatzierung beim Lesen von Datensätzen nach Aktualisierungs- und Löschvorgängen

- Verbesserte Unterstützung für das Lesen in einem DAO mit mehreren Entitäten, wenn sich die Leserichtung ändert
- Verbesserte Unterstützung für User Spaces, um zu verhindern, dass Instanzen von allen Threads wiederverwendet werden, anstatt dass jeder Thread seine eigene Instanz hat
- Verbesserte Unterstützung für gleichzeitigen Multithreading-Zugriff beim Lesen von Datensätzen
- Das Speichern des Benutzernamens/der Benutzer-ID in der YML-Konfiguration wurde verbessert SharedContext
- Die Freigabe gesperrter Datensätze mit aktualisierten Werten wurde verbessert
- Unterstützung für OPM-Compiler-spezifisches Verhalten für die Anweisung NEXT SENTENCE hinzugefügt

Transversale Fähigkeiten

Neue Features

- Die neue Eigenschaft metadata.ini wurde hinzugefügt, die den Legacy-Compiler der zu transformierenden Artefakte `legacy.compilerto` angibt. Die Unterstützung einiger COBOL-Anweisungen, wie NEXT SENTENCE, ist je nach eingestelltem Wert unterschiedlich.
  - „ZOS“ für ein z/OS-Legacy-System.
  - „FILE“ oder „OPM“ für AS4 00-Systeme. Standard = „ILE“ wenn `legacy.system` = „as400"

Verbesserungen

- Frontend — Die Bildschirmfeldkomponenten wurden neu gestaltet, um die Palette der unterstützten Feldtypen zu erweitern. Diese Verbesserung ermöglicht es der Runtime, eine größere Vielfalt an Benutzereingaben und Datenanforderungen im AS4 Zusammenhang mit 00 zu berücksichtigen.
- Verbesserte Methode `isValid()` zur Aktivierung eines separaten Vorzeichen-Bytes ZonedType
- Verbesserte Unterstützung `StringConcatenationBuilder::withPointer` für die Verkettung mit CRLF
- Die Unterstützung bestimmter Doppelbyte-Kodierungen wurde verbessert, um sie threadsicher zu machen
- Verbesserte Leistung von State Machines durch die Integration eines neuen Frameworks
- Verbesserter Algorithmus zur Zuweisungsoptimierung, um unerwartetes Umschreiben zu verhindern

## AWS Blu Age Transformation Engine 4.4.0

### zOS

#### Verbesserungen

- LISTCAT - Verbesserter Parser, um doppelte Einträge zu verhindern
- LISTCAT - Verbesserte Unterstützung von ESDS für das Dateisystem in JCL/Groovy
- CICS — Verbesserte Unterstützung für LENGTH OF für CICS-Anweisungen

### AS400

#### Verbesserungen

- Verbesserung der DDS-Datensatzgenerierung
  - Die Unterstützung von DDS-Datensätzen zur Generierung von Entitäten, die der DDS-Datensatzstruktur entsprechen, wurde verbessert
  - Es wurde Unterstützung für gemeinsam genutzte Felder und Zuordnungsfunktionen bereitgestellt, die besser zur alten Version passen
  - Die Handhabung sowohl extern beschriebener als auch programmbeschriebener Dateien wurde verbessert
- RPG — Verbesserte RPG-Erkennung für Module, bei denen nur freie Form verwendet wird
- RPG — Verbesserte Unterstützung für die COPY-Anweisung, um ein Schlüsselwort `*LIBL/` als Präfix zu ignorieren, um ein Anwendungs-Copybook zu finden
- RPG - PF - Verbesserte Unterstützung für die Eingabespezifikation mit physikalischen Datensätzen aus einer Datei
- RPG — Unterstützung für On-Exit-Statements hinzugefügt
- RPG — Verbesserte Unterstützung von Schlüsselwörtern LikeRec
- RPG — Verbesserte Zuordnung von umbenannten DSPF-Feldern
- CL - Verbesserte Auflösung von Feldnamen
- COBOL — Verbesserte Unterstützung der Konvertierung von Hexadezimal zu Zeichen
- Verbesserte Unterstützung für die Generierung von Dezimaltypen
- Verbesserte Unterstützung der FIXME-Meldung für nicht unterstützten Legacy-Code (gesamte Legacy-Zeile anzeigen)
- Verbesserte Leistung auf der AWS Transformation Engine (AS40 Analyseschritte)

- Verbesserte Unterstützung von Keyword LikeRec zur Anpassung an die Dateispezifikationen
- Verbesserte Unterstützung der integrierten Funktion %Diff
- Unterstützung für Sonderzeichen (Währungszeichen) auf dem DSPF-Etikett hinzugefügt

## Versionshinweise 4.3.0

Veröffentlichungsdatum: 16. September 2024

Diese Version von AWS Blu Age Runtime and Modernization Tools konzentriert sich auf die Erweiterung der Funktionen und des Schutzbereichs zur Modernisierung der Mainframe-Funktionen. Zu den wichtigsten Funktionen und Änderungen in dieser Version gehören:

- CICS: Zusätzliche Unterstützung für den Datenaustausch von den Terminals und die Ausführung von Transaktionen mit eingehenden Daten durch Unterstützung des Befehls SEND MAP mit Map Reference.
- JCL: Neue Funktion, die es ermöglicht, die letzte Ausführung eines Batch-Jobs von einem zuvor fehlgeschlagenen JCL/PROC-Schritt aus neu zu starten oder einen verzögerten Neustart auszulösen, indem zuvor ausgeführte Schritte umgangen werden. Dies ermöglicht eine bessere Kontrolle über die Batch-Verarbeitung mithilfe persistenter Checkpoints auf Schrittebene.
- AS400: Zusätzliche Bibliotheksunterstützung, verbesserte Leistung und Robustheit häufig verwendeter Befehle wie CPYF, OVRDBF, SBMJOB und OPNQRYF und viele mehr.

Wir haben diese Version der Blu Age Runtime mit dem folgenden Stack getestet. AWS Andere Komponentenversionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 18.18 |
| | Npm 9.8 |
| | Eckig 17 |
| Serviceschicht | Spring Boot 3.2.5 |

| | |
|---|---|
| | Spring Core 6.1.5 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 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 4.3.0

zOS

Neue Features

- CICS — Support für Kartenreferenz im Befehl SEND MAP hinzugefügt
- CICS — Unterstützung für den Befehl RECEIVE und Unterstützung für die Ausführung von Transaktionen mit Daten vom Bildschirm hinzugefügt `JicsTransactionRunner`
- Unterstützung für den IIH-Header für die JMS-Nachrichten hinzugefügt
- COBOL — Unterstützung für mehrere eingebettete Leerzeichen in Pseudotext für die REPLACING-Anweisung hinzugefügt
- COBOL — Unterstützung für die JSON-PARSE-Anweisung hinzugefügt
- Blusam — Unterstützung für KMS zur Funktion „Datensatz exportieren“ hinzugefügt
- BAC — Die Konfiguration `application-main.yaml` zur Definition der Datensatzgröße wurde hinzugefügt, um geladene Masken zu filtern, die dieser Datensatzgröße entsprechen
- JCL - INFUTILB - Unterstützung für das Schlüsselwort INTO als Teil der BMC-Steueranweisung hinzugefügt
- GS21 - SOSI-Behandlung für JEF-Kodierung hinzugefügt
- GS21 - JCL - Als KDJBR14 Alias für hinzugefügt IEFBR14
- GS21 - JCL - KQCAMS wurde als Alias für IDCAMS hinzugefügt
- MF — Unterstützung für COBOL MF-kompatible Dateien hinzugefügt, abhängig von der Feldunterstützung

- MF — Unterstützung für den SORT-Mechanismus für COBOL MF-kompatible Dateien hinzugefügt
- MF — Unterstützung für COBOL MF Compatible zum Öffnen nicht optionaler fehlender Dateien hinzugefügt

Verbesserungen

- JCL - DSNUTILB - Verbesserter LOAD-Betrieb mit ZONED DECIMAL Type
- JCL - DSNUTILB - Unterstützung für doppelte Schlüssel hinzugefügt
- JCL - DSNUTILB - Unterstützung für den Rollback-Mechanismus beim LOAD-Befehl hinzugefügt
- JCL - INFUTILB - Verbessertes UNLOAD mit den neuen Eigenschaften FETCHSIZE und CHUNKSIZE
- JCL - A - IKJEFT1 Verbessertes Lesen von SYSTSIN-Dateien durch Hinzufügen des aktuellen Zeichensatzes
- JCL - DFSORT - Unterstützung für Option & hinzugefügt DATE4 DATE5
- JCL - DFSORT - Unterstützung für den Fall des Typs „Variabler Block“ als Eingabe und des Typs „fester Block“ als Ausgang hinzugefügt
- JCL - DFSORT - Unterstützung für ALTSEQ hinzugefügt
- JCL — Verbesserte Checkpoint-Metadaten mit Job-Web-ID
- JCL — Verbesserte Batch-Neustart-Checkpoint-Bereinigung für REDIS
- IMS — Die EXPRESS-Funktion für den PURGE-Befehl wurde implementiert
- IMS — Unterstützung für die Optionen PCBNAME und LIST für PCB-Anweisung hinzugefügt
- COBOL — Unterstützung für GO TO-Anweisungen ohne Ziel hinzugefügt
- CICS — Verbesserte Unterstützung für INTO-Anweisungen in READQ RecordAdaptable TS
- CICS — Verbesserte Unterstützung für den Befehl INQUIRE TRANSACTION
- CICS — Verbesserte Unterstützung für SetBytes im Befehl READNEXT
- CICS — Verbesserte Unterstützung für den START-Befehl ohne CHANNEL-Option
- CICS — Support für den Referenztyp für Anfragen hinzugefügt TSQueue
- CICS — Verbesserte Unterstützung für den Befehl RECEIVE MAP, wenn Map und Mapset Reference sind
- CICS — Verbesserte Unterstützung für die Optionen FROM und LENGTH für den Befehl RECEIVE MAP

- CICS — Unterstützung für Attribute hinzugefügt RecordAdaptable
- CICS — Verbesserte Unterstützung für den Befehl RECEIVE zur Behandlung von Überläufen
- CICS — Unterstützung für Slice-Regeln in CICS-Anweisungen hinzugefügt
- CICS — Verbesserte Unterstützung für die Linkage-Strukturen DFHCOMMAREA und DFHEIBLK. Die Transformationsengine unterstützt mehr implizite Definitionen
- CICS — Unterstützung für die Optionen START, NEXT und END für den Befehl INQUIRE CONNECTION hinzugefügt
- CICS — Unterstützung sowohl für den Typ 'int' als auch für 'reference' für die Option LENGTH des Befehls RECEIVE hinzugefügt
- CICS — Verbesserte Unterstützung für das Parsen des INQUIRE NETNAME-Befehls
- CICS — Unterstützung für Gruppennamen hinzugefügt für JicsQueueBuilder
- Blusam — Unterstützung für indizierte Dateien hinzugefügt, die mit einem generischen Schlüssel beginnen
- Blusam - Verbesserte Blusam-Loader
- BAC — Verbesserte Unterstützung für die Datensynchronisierung in Umgebungen mit mehreren Instanzen, wenn Redis zur Zentralisierung zwischengespeicherter Werte verwendet wird, einschließlich der tatsächlichen Daten und Sperren
- BAC — Verbesserte Benutzeroberfläche (Stil, Logo, Checkbox)
- BAC und JAC — Die Konfiguration von `application-main.yaml` wurde hinzugefügt, um den Benutzernamen und das Passwort des standardmäßigen Super-Admin-Benutzers im Secret von AWS Secrets Manager abzurufen, indem der ARN angegeben wird
- BAC und JAC — Aktualisierung der Abhängigkeit auf Bootstrap 5
- Verbesserte JCL-Checkpoints und die JICS Redis-Vorlagenkonfiguration TSQueues
- Verbesserte Unterstützung für Size of Pointer, abhängig von AMode
- Unterstützung für Nullvergleiche bei hinzugefügt NumericEditedType
- Vor der Protokollierung wurden die SLF4j-MDC-Eigenschaften erzwungen
- Die Unterstützung für das Lesen von Dateien wurde verbessert, sodass mehrere Leerzeilen verarbeitet werden können
- MF — Verbesserte Unterstützung für die Initialisierung von Zeigervariablen für die COBOL MF-Compilerdirektive InitPtr
- Redis — Verbesserte Funktion zum Parallelitätsaspekt durch eine auf GwFileLock Redisson basierende Implementierung

AS400

Neue Features

- CL - Unterstützung für den CHGPF-Befehl hinzugefügt
- RPG - Unterstützung für die Funktionen %HOURS, %MINUTES und %SECONDS hinzugefügt
- COBOL - Unterstützung für SORT-Dateien mit der Blu4IV DAO-Architektur hinzugefügt

Verbesserungen

- CL — Es wurde verbessert PgmClose , um als Programm registriert zu werden und eine Vielzahl von Objekten für den OPNID-Parameter zu akzeptieren
- CL — RTVMBRD wurde überarbeitet, um mehrere Bibliotheken und Mitglieder zu verwalten
- CL — Unterstützung für den Parameter TOLIB im MOVOBJ-Befehl hinzugefügt
- CL — Verbesserte Unterstützung der Partition beim Befehl CPYFRMSTMF
- CL — Unterstützung für den SNDMSG-Parameter TOUSR hinzugefügt
- CL - Verbesserte Unterstützung des OVRDBF-Befehls
- CL — Verbesserte Leistung für den Befehl OVRDBF — Aktualisieren Sie die Standardwerte für srcfile und member
- CL — Verbessertes Kopieren von Dateien mit dem Befehl CPYF
- CL — Der CPYF-Befehl wurde überarbeitet, um robuster zu sein und QTEMP, CRTFILE, FROMRCD & TORCD, MBROPT und FMTOPT (MAP & DROP) besser zu handhaben
- CL — Verbesserte Unterstützung für den CPYF-Befehl in Fällen, in denen FROMFILE und TOFILE nicht übereinstimmende Spalten haben
- CL — Die Behandlung von Spalten mit unterschiedlichen Namen in CPYF NOCHK wurde verbessert, wenn REPLACE angegeben ist
- CL — Leere Implementierung für den Befehl CRTDUPOBJ für logische Dateien hinzugefügt
- CL — Das Problem mit der Substring-Indizierung wurde mit dem Befehl CHGDTAARA behoben
- CL — Verbesserte Unterstützung des Befehls SBMJOB
- CL — Groß- und Kleinschreibung wird bei der OverrideManager Erstellung und OpnqryfHelper Zuordnung nicht berücksichtigt
- Bildschirm — Der anfängliche Fokus des ersten editierbaren Felds, wenn kein Cursor angegeben ist, wurde verbessert

- Bildschirm — Verbesserte Fokusposition nach dem Schließen und bei Verwendung des Hilfemenüs
- Bildschirm — Verbesserter Cursorfokus nach dem Drücken der Seite hoch/runter in der Tabellenkomponente
- Bildschirm — Verbesserte Unterstützung für mehrere Feldfehlermeldungen und Fokus
- Bildschirm — Verbesserte Berechnung der Zeilennummer für Felder in Unterdateien
- Bildschirm — Verbesserte Unterstützung von Unterdateien, die mit SFLINZ initialisiert wurden
- Bildschirm — Die Unterstützung für rein numerische Eingaben wurde verbessert
- Bildschirm — Verbesserte Behandlung des WINDOW-Schlüsselworts in DSPF mit 3 Parametern
- Bildschirm — Verbesserte Position der Fußzeile für Tabellen mit Datensätzen, die mehr als eine Zeile enthalten
- Bildschirm — Die Seitennavigation wurde verbessert, sodass die Nachricht beim Wechseln zwischen Bild hoch und runter haften bleibt
- Verbesserte EDITC-Funktionalität für Bearbeitungscode 3
- Der Blu4iv-Sperrmechanismus für den Datenbereich wurde verbessert, sodass nichts unternommen wird, wenn keine Sperre entsperrt werden muss, anstatt eine Ausnahme auszulösen
- Es wurde Unterstützung hinzugefügt, um die Anzahl der betroffenen Zeilen zurückzugeben in StraightQueryBuilder
- Verbesserter QTEMP-Protokollmechanismus
- Verbessert DAOManager reads/writes/deletes für Anwendungsfälle bei Dateien, die von einer anderen File+-Bibliothek überschrieben wurden

Transversale Funktionen

Neue Features

- Es wurde eine zentrale Möglichkeit hinzugefügt, die SSL/TLS-bezogenen Systemeigenschaften nach Konfiguration zu verwalten. Dies ermöglicht die Verwendung von AWS Secrets Manager
- Verbesserte Konfiguration von IBM MQ-Ressourcen mit AWS Secrets Manager
- JCL — Es wurde die Konfiguration für temporäre Speicherorte für Runtime hinzugefügt, bei der Groovy-Dateien mithilfe der Eigenschaft YML aufgelöst wurden. Außerdem wurde die Möglichkeit hinzugefügt, mithilfe der Eigenschaft tempFilesDirectory YML anzugeben, ob der Inhalt des Ordners für temporäre Dateien beim Start der Anwendung gelöscht werden soll cleanTempFiles DirectoryAtStartup

- AWS-Geheimnisse für alle Redis-Anmeldeinformationen hinzufügen

Verbesserungen

- Verbesserte Konvertierung vom Typ alphanumerisch zum Typ numerisch bearbeitet
- Verbesserte DataUtils: :isNumeric Prüfung für PackedType
- Verbesserter Zeitstempel für Protokolldateien
- Separate Anmeldung abgewickelt. ZonedType decodeAsString
- COBOL - Verbesserte Unterstützung der Anweisung INITIALIZE
- Verbesserte Unterstützung von. DataUtils compareAlphInt zur Verarbeitung von führenden und abschließenden Leerzeichen für AS4 00 und ZOS
- SQL — Verbesserte implizite Cursor-Laufzeitvalidierung mit Nur-Lese-Zugriff
- SQL — Verbesserter Mechanismus zum Zwischenspeichern von Metadaten
- Entfernen Sie die Jics/Blusam-Datenbankverbindung aus der Gapwalk-Anwendung `application-main.yml`

## Version 4.3.0 der Modernisierungstools

zOS

Neue Features

- GS21 - Unterstützung für COBOL GS21 CONSTANT SECTION hinzugefügt
- GS21 - JEF-Kodierung zu verfügbaren Zeichensätzen hinzugefügt

Verbesserungen

- CICS — Unterstützung für das Parsen des Befehls DOCUMENT CREATE hinzugefügt
- CICS — Unterstützung für das Parsen des CICS WEB EXTRACT-Befehls hinzugefügt
- CICS — Unterstützung für das Parsen des Befehls WEB WRITE hinzugefügt
- CICS — Transformationsunterstützung für DB2 CONN SIGNIN und PLAN hinzugefügt
- CICS — Verbesserte Unterstützung für das Parsen des SEND MAP-Befehls durch Ignorieren der Option TERMINAL

- CICS — Verbesserte Unterstützung für das Parsen des RETURN-Befehls durch Ignorieren der Option ENDACTIVITY
- MFS — Verbesserte Unterstützung für die Generierung von MFS-Dateien mit einer bestimmten Erweiterung
- COBOL — Verbesserte Unterstützung für die REPLACE-Anweisung
- COBOL — Behandelte dynamische Pfade und MF-Compiler-Direktiven
- COBOL — Verbessert die Unterstützung für den Wert WEGGELASSEN in der CALL-Anweisung
- COBOL — Verbesserter Zugriff auf mehrdimensionale Felder zur Unterstützung von vorzeichenbehafteten Werten
- COBOL — Unterstützung für die Klausel OF für die Anweisung FILE STATUS hinzugefügt
- COBOL — Verbesserte Analyse der Anweisung RESULT-SET-LOCATOR
- JCL - IDCAMS - Unterstützung für die Abkürzung RECORDS hinzugefügt

AS400

Neue Features

- CL - Unterstützung für zeigerbasierte und definierte Variablen in der CL-Transformation hinzugefügt
- CL — Unterstützung für Sonderzeichen in DCLF hinzugefügt
- Unterstützung für die Retrieve Call Stack (QWVRCSTK) API wurde hinzugefügt

Verbesserungen

- RPG - Verbesserte Transformation von Verfahrensparametern mithilfe `likeds` von Schlüsselwörtern
- RPG — Überprüfen Sie die Unterstützung des Schlüsselworts EXTNAME
- RPG — Der Literalwert *ALL wurde für die Unterstützung verbessert
- RPG — Verbesserte Unterstützung für Ausgabespezifikationen und Dateien mit Programmbeschreibungen
- DDS — Verbesserte Auflösung von DDS-Feldern in einem LF, der auf einen PF verweist, der auf einen Wörterbuch-PF verweist
- Bildschirm — Die Anzeigen wurden gelöscht, wenn die CLEAR-Anweisung verwendet wurde, um einen Datensatz aus dem DSPF zu löschen

- CL — Verbesserte Transformation/Generierung von CL-Parametern mit Elementlisten

Transversale Fähigkeiten

Verbesserungen

- SQL — Verbesserte Generierung von SQL-Abfragen, die N mit Tilde-Zeichen enthalten
- COBOL — Verbesserte Unterstützung der Anweisung LENGTH OF für Gruppenfelder
- COBOL — Verbesserte Unterstützung von REDEFINED-Feldern mithilfe von Copybooks

## Versionshinweise 4.2.0

Veröffentlichungsdatum: 10. Juli 2024

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

- Wir haben die Transformationsleistung verbessert, insbesondere bei großen Projekten mit über 30 Millionen Codezeilen. Wir haben eine Reihe von Verbesserungen eingeführt, und die erzielten Ergebnisse zeigten eine Zeitersparnis von über 150% und Läufe, die in Minuten statt in Stunden abgeschlossen wurden. Die wichtigste Verbesserung, die wir eingeführt haben, ist die Konfiguration eines Timeout-Mechanismus, der die maximale Analysezeit begrenzt, sodass Dateien mit erkannten Problemen übersprungen werden. Wir markieren übersprungene Dateien, sodass Sie sie bei Bedarf später untersuchen können.
- Wir haben Unterstützung für ein verteiltes Sperrverwaltungssystem für AS4 00 Projekte hinzugefügt. In einer Hochverfügbarkeitsumgebung (mehrere Knoten), in der mehrere Instanzen der Anwendung auf dieselbe Datenbank abzielen, stellt die Aufrechterhaltung der Datenkonsistenz während des gesamten Lebenszyklus dieser Instanzen eine erhebliche Herausforderung dar. Um dieser Herausforderung effektiv zu begegnen, haben wir Redis als gemeinsam genutzten und externen Caching-Server hinzugefügt, der bei der Ausführung im Batch-Modus die Koordination zwischen allen Instanzen übernimmt.
- Wir haben eine neue dynamische Paginierungsfunktion für die Tabellenkomponente hinzugefügt. Ziel dieser Funktion ist es, die Antwortzeit zu verbessern und den Speicherverbrauch für Tabellen mit einer großen Anzahl von Zeilen zu reduzieren. Diese Funktion ermöglicht es der Tabellenkomponente, nur einen Teil der Daten zu laden und bei Bedarf mehr Datensätze abzurufen, während Sie durch die Seiten navigieren. Um die Benutzererfahrung weiter zu

verbessern, unterstützt die Plattform auch das Vorabrufen von Daten. Diese neue dynamische Paginierungsfunktion bietet eine effizientere und reaktionsschnellere Benutzererfahrung für Anwendungen mit großen Datensätzen.

- Um einer wichtigen Herausforderung zu begegnen, die häufig auftritt, haben wir Unterstützung für verschachtelte COBOL-Programme hinzugefügt. Bisher bestand die Problemumgehung für die Modernisierung verschachtelter COBOL-Programme darin, Programme manuell in verschiedene Dateien aufzuteilen, sie über den Linkage-Bereich zu verknüpfen und sie gegenseitig mit den erforderlichen Argumenten aufrufen zu lassen. Dieser Prozess war nicht nur zeitaufwändig, sondern auch fehleranfällig. Sie können jetzt verschachtelte COBOL-Programme modernisieren, ohne dass eine manuelle Trennung erforderlich ist.

Wir haben diese Version der AWS Blu Age Runtime mit dem folgenden Stack getestet. Andere Komponentenversionen sind möglicherweise ebenfalls kompatibel.

| Komponente | Version getestet |
| --- | --- |
| Java | Java 17 |
| Präsentationsebene | Knoten JS 18.18 |
| | Npm 9.8 |
| | Eckig 17 |
| Serviceschicht | Spring Boot 3.2.4 |
| | Spring Core 6.1.5 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 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 4.2.0

zOS

Neue Features

- DB2 - Unterstützung für den Aufruf von gespeicherten Prozeduren ohne Schemakennzeichner in der SQL-Abfrage hinzugefügt
- COBOL — Unterstützung für die HEX-OF-Funktion hinzugefügt
- COBOL — Unterstützung für verschachtelte Programme hinzugefügt
- COBOL — Unterstützung für FUNCTION hinzugefügt und TEST-DATE-YYYYMMDD TEST-DAY-YYYYDDD
- CICS — Unterstützung für die Option UCTRANST im Befehl SET TERMINAL hinzugefügt
- CICS — Unterstützung für den Befehl INQUIRE CONN wurde hinzugefügt DB2
- BluSam - Unterstützung für das Löschen von Schlüsseln bei dynamisch aufgerufenem VSAM hinzugefügt
- IMS — Unterstützung für den Befehl TERM wurde hinzugefügt
- BAC — Autorisierungsprüfungen für alle BAC-REST-Endpunkte hinzugefügt
- BAC — Konfiguration hinzugefügt, um eine Datensatzgröße `application-main.yaml` zu definieren, um geladene Masken zu filtern, die dieser Datensatzgröße entsprechen
- BAC und JAC: Konfiguration hinzugefügt, `application-main.yaml` um den Benutzernamen und das Passwort des Standard-Super-Admin-Benutzers im Geheimen abzurufen, `command` indem der ARN angegeben wird

Verbesserungen

- JCL - SORT - Verbesserte Unterstützung für die OMIT-Klausel zur Behandlung von Bedingungen mit Shiftin und Zeichen ShiftOut
- JCL - SORT - Verbesserte Unterstützung für das BDW-Feld
- JCL - SORT - Verbesserte Unterstützung für mehrere GDG-Verkettungen mit dem BDW-Feld
- JCL - DFSORT - Unterstützung für INREC PARSE STARTAFT//STARTAT-Klauseln hinzugefügt
- JCL - IEBGENER - Verbesserte RecordSize-Behandlung für Ausgabedateien
- JCL - INFUTILB - Der NULL-INDIKATOR wurde auf Basis von YML deaktiviert. GRAFISCHER FALL KORRIGIEREN

- JCL — Verbesserte Unterstützung für die Behandlung von Konstanten im FormatterParser OUTREC-Feld
- JCL — Verbesserte Ladedaten für Grafiktypen im DSNUTILB-Programmdienstprogramm
- JCL - SORT - Verbesserte Unterstützung für das Zoned Decimal-Format
- JCL - SORT - Verbesserte Unterstützung für die OMIT-Klausel zur Behandlung von Bedingungen mit Shiftin und Zeichen ShiftOut
- MQ — Die Handhabung der MQ-Verbindung wurde verbessert, um sie an mehrere Geschäftsabläufe anzupassen
- CICS — Verbesserte Unterstützung der Zeigerreferenz für EXEC CICS READ SET (ptr-ref) - Anweisungen
- COBOL — Verbesserte Unterstützung für ADDRESS OF Linkage-Abschnittsdatensätze
- COBOL — Unterstützung für EXP- und 0-Funktionen hinzugefügt EXP1
- COBOL — Verbesserte Unterstützung für die REPLACE-Anweisung mithilfe von Copybook
- COBOL — Verbesserter mehrdimensionaler Feldzugriff zur Unterstützung von vorzeichenbehafteten Werten
- MF COBOL - Unterstützung für sequentielle Dateien mit variablem Format hinzugefügt
- IMS — Verbessertes Lesen der Konfiguration von IMS-YML-Dateien, um die Verwendung von Umgebungsvariablen zu ermöglichen
- IMS — Es wurden zusätzliche Möglichkeiten zur Angabe der Segmentnummer behandelt
- IMS — Die Stabilität wurde erhöht, wenn ein IMS-Programm von einer programmgesteuert gestarteten Transaktion aus aufgerufen wird
- IMS — Die Suchkriterien wurden verbessert, sodass die aktuelle Länge der WHERE-Klausel berücksichtigt wird, falls die implizite Segmentlänge nicht angegeben wird
- IMS — Verbessertes Lesen der Konfiguration von IMS-YML-Dateien, um die Verwendung von Umgebungsvariablen zu ermöglichen
- Verbesserte Unterstützung für die VALUE-Klausel in NumericEditedType
- Verbesserte Unterstützung für die Verkettung von Zeichenketten für den Fall, dass die erste zu verkettende Zeichenfolge leer, leer oder Leerzeichen ist

AS400

Neue Features

- Unterstützung für Seitennummerierung innerhalb der Tabellenkomponente hinzugefügt. Projekte können diese Funktion verwenden, um die Antwortzeit und -größe zu verringern, wenn eine Tabellenkomponente mit einer großen Anzahl von Zeilen geladen wird
- Bibliotheksunterstützung für SQL-Abfragen in der AS4 00-Anwendung hinzugefügt. Da Bibliotheken in modernen Anwendungen in Partitionen konvertiert werden, haben wir die Laufzeit angepasst, um die Abfragen entsprechend neu zu schreiben
- RPG — Unterstützung für die QTEMP-Bibliothek für SQL-Abfragen hinzugefügt
- RPG — Der CONVERT-Funktion wurde eine Kodierung hinzugefügt, um leere Eingabewerte zu verarbeiten
- RPG — Unterstützung für die Funktionen %HOURS, %MINUTES und %SECONDS wurde hinzugefügt
- CL — Der Befehl CHGPFM wurde hinzugefügt
- CL — Unterstützung für das Schlüsselwort *FROMLIB im Befehl CRTDUPOBJ hinzugefügt
- CL — Unterstützung für die Erstellung von Tabellen und Partitionen für Tabellennamen mit mehr als 9 Zeichen hinzugefügt
- CL — Unterstützung für das Löschen von Flatfiles in Unterordnern für den DLTF-Befehl hinzugefügt

Verbesserungen

- Bildschirm — Es wurde verbessert ErrorMessage , um es an ein bestimmtes Feld zu binden und es hinzuzufügen ArrayMessageLine
- Bildschirm — Verbesserter errormsg-Cursor
- Bildschirm — Verbessert ArrayMessageLine , sodass er jetzt nicht mehr in der Tab-Reihenfolge enthalten ist
- Bildschirm — Verbesserte Anzeige von Fehlermeldungsarrays für den AS4 00-Bildschirm
- SQL — Verbesserte Cursorunterstützung für die Übertragung der Transaktion beim Schließen, um Deadlocks bei der Partitionserstellung zu vermeiden
- CL — Unterstützung für den PgmCall Befehl hinzugefügt und das nicht unterstützte QCMDEXC-Muster verbessert
- CL — Verbesserte Unterstützung für den Befehl CHKOBJ zur Verarbeitung von OBJTYPE PGM
- CL — Verbesserte Unterstützung mehrerer Bibliotheken für CPYF und andere CL-Befehle, die sich mit Bibliotheken und Partitionen befassen

- CL — Unterstützung für die Übergabe einer Programmnamenvariablen im CALL PGM-Befehl hinzugefügt
- CL — Hat den Fall für den Standardtyp des Objekttyps behandelt
- CL — Unterstützung mehrerer Bibliotheken für den Befehl CRTDUPOBJ hinzugefügt
- CL — Verbesserte Handhabung von Datenbankverbindungen bei mehreren Befehlen
- CL — Verbesserte Unterstützung für RMVLNK für den Fall, dass eine Datei oder ein Verzeichnis nicht gefunden wurde und die CPF0000-Monitormeldung angezeigt wird
- CL — CLRPFM wurde verbessert, um die Bibliothek beim Entfernen von Datensätzen zu berücksichtigen
- CL - CPYF - Verbesserter Befehl zur Unterstützung der QTEMP-Bibliothek, des Parameters FmtOpt (*) und des Steuerzeichens NoChk
- CL — Die Behandlung von Anführungszeichen und fehlenden Parametern in den Befehlen RMVLNK und CPY wurde korrigiert
- RPG — Verbessertes Variablen-Scoping; DataArea befindet sich jetzt im funktionierenden Bereich statt im Linkage-Bereich
- RPG — DAO-Leseabfragen wurden verbessert, sodass sie jetzt auch ohne Transaktion ausgeführt werden können, um Deadlocks zu vermeiden
- Verbesserte MQ-Messaging-Suche durch Hinzufügen einer Kürzung zu MSGQ bei DB-Suche
- Unnötige Transaktionsdeklarationen zur Unterstützung von Datenbankverbindungen wurden entfernt
- Die Aktualisierung des Quartz-Jobstatus im Ausnahmefall wurde verbessert
- Unterstützung für den Fall hinzugefügt, dass ein Indikator-Array nicht initialisiert ist

Transversale Fähigkeiten

Neue Features

- Redis — Die globale Redis-Konfiguration wurde für alle Redis-Caches hinzugefügt
- Es wurde eine Funktion zur Sitzungsverfolgung hinzugefügt, um das Speichern von Sitzungsverfolgungsinformationen (Sitzungs-ID, zugehöriger Benutzername, Erstellungszeitstempel und Knoten-ID) durch persistente Speicherung der Daten in Redis zu ermöglichen
- Es wurde eine temporäre Speicherortkonfiguration für zur Laufzeit aufgelöste Groovy-Dateien über die YML-Eigenschaft hinzugefügt. Außerdem wurde die Möglichkeit

hinzugefügt`tempFilesDirectory`, mithilfe der YML-Eigenschaft anzugeben, ob der Inhalt des Ordners für temporäre Dateien beim Start der Anwendung gelöscht werden soll `cleanTempFilesDirectoryAtStartup`

Verbesserungen

- Verbesserte Unterstützung für die Konfigurationseigenschaften der Implementierung von Verbindungspools für Versorgungsdatenquellen
- Verbesserte Unterstützung für den Druckermodus und die ANSI-Laufwerkssteuerung auf der Grundlage der Verwendung von ADVANCEDING-Klauseln und WRITE BEFORE-Klauseln
- Aktualisierte Angular-Version in der Frontend-Anwendung für modernisierte Projekte
- Verbesserte Secret Manager-URL-Syntaxkonstruktion für DB2
- Verbessert die DataUtils. compareAlphInt Methode, um Unterstützung für nachfolgende Leerzeichen hinzuzufügen
- Verbesserte SQL-Unterstützung für die Ausgabe vom Typ Blob
- Robustheit für Job-Trigger über den Post/Script-Endpunkt hinzugefügt

## Version 4.2.0 der Modernisierungstools

zOS

Neue Features

- CICS — Unterstützung für das Parsen von WEB-CICS-Befehlen hinzugefügt
- CICS — Unterstützung für die Transformation des MONITOR-Befehls hinzugefügt
- CICS — Unterstützung für das Parsen des CICS-Befehls SEND MRO hinzugefügt
- COBOL — Unterstützung für das Parsen der NO REWIND-Anweisung hinzugefügt
- COBOL — Unterstützung für den Zahlentyp der Option UCTRANST im CICS-Befehl SET TERMINAL hinzugefügt
- COBOL — Unterstützt die MULTIPLE FILE-Klausel in I-O-SECTION
- CSD — Unterstützung für die Transformation mehrerer CSD-Dateien hinzugefügt
- CSD — Unterstützung für die Generierung jicsFileAix von.json aus mehreren CSD-Dateien hinzugefügt
- IDCAMS — Unterstützung für die Erstellung eines relativen Datensatzes (RRDS) hinzugefügt

### Verbesserungen

- Verbesserte Leistung bei der Berechnung von SQL-Masken
- COBOL - Verbessertes Parsen der nutzlosen RESERVE-Klausel in FILE-CONTROL
- COBOL - Verbessertes Parsen von SECTION und CLASS
- COBOL — Verbesserte DFHRESP-Behandlung
- COBOL — Verbesserte Unterstützung für EXIT PARAGRAPH durch Perform
- IMS — Verbesserte Unterstützung für Segmentnamen, die mit doppelten Klammern angegeben werden
- IMS — Die Generierung von Statuscodes wurde erweitert, wenn SCHD und TERM aufgerufen werden
- COBOL — Verbesserte Generierung von DEPENDENING ON-Feldern
- COBOL — Verbesserte Transformation der eingebauten Funktion DB2 TO_TIMESTAMP

## AS400

### Neue Features

- Unterstützung für die Konvertierung alphanumerischer Felder als CHAR in SQL-Skripten hinzugefügt
- COBOL400 — Unterstützung für vom Programm beschriebene DATABASE-Dateien hinzugefügt

### Verbesserungen

- DDS — Verbesserte Unterstützung für ALIAS-Namen
- Verbesserte Unterstützung für den Typ Float ohne Anfangswert
- COBOL 400 — Verbesserte Größenberechnung für Typen mit Vorzeichen in Zonen

## Transversale Fähigkeiten

### Verbesserungen

- Verbesserte Fehler-ID-Berichterstattung rund um DDS und SQL-Parsing
- Verbesserte Codegenerierung für Bedingungszweige

- Verbesserte Leistung bei der Generierung von Wetterberichten

## Versionshinweise 4.1.0

Veröffentlichungsdatum: 31. Mai 2024

Diese Version der AWS Blu Age Runtime and Modernization Tools konzentriert sich auf Leistung und Sicherheit. Einige der wichtigsten 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 hinzugefügt OAuth, der mehr Flexibilität bietet, sodass Kunden den Identitätsanbieter ihrer Wahl verwenden können.
  - Die Sicherheitsfunktionen wurden mithilfe geeigneter Header konfiguriert und erweitert.
- AS400 — Unterstützung mehrerer Knoten für den Datenbanksperrmechanismus. 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 AWS Mainframe-Modernisierung.

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<br>Npm 9.8 |

| | |
|---|---|
| | Eckig 16.1 |
| Serviceschicht | Spring Boot 3.2.5 |
| | Spring Core 6.1.5 |
| | Spring Statemachine 4.0.0 |
| Persistenzschicht | PostgreSQL-Engine 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 dynamische OAuth2 Provider-Handhabung 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 wurde hinzugefügt AWS Secrets Manager , sodass Sie ein SSL-Zertifikat (sslTrustStoreLocation) und ein Passwort (sslTrustStorePassword) 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 jics XAPrepare () 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 Kundenseite 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
- 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 AS4 00-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 so verbessert, dass es CPF24 07 sendet und CPF2419 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 ClCommand 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/-Transformation wurde optimiert. PL1
- 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

Veröffentlichungsdatum: 8. April 2024

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 zur 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 AS4 00 Online-Anwendung zur Ausführung von 00-Anwendungen mit Parallelisierung. AS4
  - 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 der Komplexität älterer Systeme innewohnen.

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 OUTFIL NOMATCH- (..).
- SORT — Unterstützung für die SYMNAMES-Datendefinition hinzugefügt.

- SORT — Verbesserte Behandlung der Argumente TO= und LENGTH=.
- SORT — Verbesserte Handhabung bei der MOD-Disposition.
- 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 Abfragen. DB2
- Unterstützung für Spring Beans hinzugefügt.
- Verbessert SQLConverter , 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 sich um einen Zeichensatz von 0 handelt. CUSTOM93
- 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 Anweisung TRANSFORM hinzugefügt.
- COBOL — Unterstützung für 9 neue mathematische Funktionen hinzugefügt.
- COBOL — Verbesserte Unterstützung für INTEGER-OF-DAY FUNCTION.
- COBOL — Verbesserte Unterstützung für Level 88 mit figurativem Wert.
- 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 RCDDLM 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 nun 07 gesendet wird und wenn Datei/ID nicht gefunden wurde. CPF24 CPF2419
- 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 Benutzerbereiche QUSRUSAT und QUSCUSAT hinzugefügt. APIs
- 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 das Feld als Standardtabellennamen abzurufen.

- CL - RTVJOBA Unverfügbarer Wert für den Parameter: USRLIBL.
- CL — Führende Schrägstriche im MSGF-Parameter SNDPGMMSG wurden verarbeitet.
- CL — Verbesserte Unterstützung für Platzhalter in der Quelldatei im Befehl DSPFFD.
- CL — Verbesserte Handhabung des Parameters PGMQ in RCVMSG und SNDPGMMSG.
- 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 BigDecimal Verwendung zu verhindern.

## 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 für Felder hinzugefügt. ALPHANUMERIC/ALPHABETIC/GRAPHIC

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 Mausrad aktiviert.
- Unterstützung für Funktionstasten für den Hilfescreen hinzugefügt.
- Verbesserte Unterstützung für langen Text in der split-dynamic-field Komponente.
- Verbesserte Handhabung von LF-Dateien mit mehreren Datensätzen beim Umbenennen von Datensätzen.
- CL - Verbesserter RTVJOBD-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 Anweisungen folgen. IFxx
- RPG — Unterstützung für Felder hinzugefügt, die im Prototyp mit LIKE definiert wurden.
- RPG — Die Unterstützung für das Schlüsselwort LIKE in einem von LIKEREC erstellten Feld wurde verbessert.

- RPG — Verbesserte Generierung von Operatoren mit figurativen Elementen.
- RPG - Das Parsen für den Array-Ausdruck 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 die 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 DB2 SET-Anweisung beteiligt sind.
- Verbesserte Fehlerauslösung am Ende des abgerufenen Cursors, wenn es sich bei der Ausgabe um ein einzelnes Entitätsarray 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 kann ByteRange jetzt überprüft werden, ob kein Wert mit falschen Werten erstellt wurde. byteRangeBoundsPrüfen

- removeSOSI () wurde verbessert, um die Initialisierung von a GraphicAlphanumericType mit einem leeren Zeichen zu unterstützen
- 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 Datensatzsperre
- 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 verwandte Bereiche hinzugefügt APIs
- 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 wurde verbessert DataUtils. compareAlphInt() wenn es um Polsterung 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 verringern (CVE-2024-21634, CVE-2023-34055, CVE-2023-34462, -JAVA-ORG/SPRINGFRAMEWORKSECURITY-5905484, CVE-2023-46120, CVE-2023-6481, CVE-2023-6378, CVE-2023-5072) IN1

## 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 Control Level Indicator 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 — Verbesserter Umgang mit Gruppierungstrennzeichen 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 Verarbeitung von Prozeduraufrufen innerhalb von Bedingungen IF/ELSEIF/WHEN
- 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 in AS4 00 Unterstützung für jobübergreifende Funktionen hinzu, wobei der Hauptvorteil darin liegt, dass Jobs Anfragenachrichten synchron zwischen ihnen versenden 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 verketteter sequentieller 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/unsignierte) 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 .chargraphic wurde korrigiert, um SOSI aus einem Grafiktyp zu entfernen
- Leistung verbessern auf. DataUtils isDoubleByteKodierung
- 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 Datensatzsperre (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

AS400

Neue Features

- Erstellt eine Registrierung im Anwendungskontext
- Support für das DSPF-Schlüsselwort CLRL (NO) Support die Überwachung von Datensatzsperren
- Support für Keyed DataQueue
- Support für INQUIRY-Nachrichten für Batch-Jobs
- Unterstützung für die vom Programm beschriebene Druckerdatei für AS4 00 COBOL hinzugefügt

- Verarbeitet den Befehl RMVJOBSCDE cl
- Verbesserung für RUNSQL/DLYJOB
- CHKOBJ: Es wird ein älterer Fehlercode für den Parameter LIB ausgelöst
- SNDPGMMSG: 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
- SNDPGMMSG 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.netfective.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 beim Umgang mit 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-Objekttyp
- 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-Verpackungen hinzugefügt EC2
- 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 Zonendezimalwerten
- COBOL: Fügt Unterstützung für das Escapen von einfachen/doppelten Anführungszeichen in Zeichenkettenliteralen hinzu

### AS400

#### Verbesserungen

- Die Verarbeitung des Bearbeitungscodes 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

- Die Handhabung von Indikatoren für Kontoauszüge wurde verbessert CABxx
- 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
- Behandelter 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: ebcdicFilesWith VarcharIn VB-Eigenschaft 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 COBOL2 Model, RPGCycle Builder, 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 Funktionen APIs sind verfügbar, um den Betrieb und die Interaktion mit BluSAM-Datensätzen zu verbessern, wie z. B. der integrierte Import in den Managed Service und die Möglichkeit, Datensatz-Metadateninformationen aufzulisten.
- Leistungsverbesserungen und Erweiterung der Nutzung von Redis, einschließlich Verfügbarkeit im Clustermodus, Abruf hochverfügbarer Daten 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 bei der Grammatikinitialisierung von OUTFIL TRAILER1 + DFSORT hinzugefügt
- Gemeinsames DDUtils 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 DSNTEP und INFUTILB. BLOB wurden jetzt DB2 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
- Verbesserter Umgang mit Datenbankverbindungen im SQL-Framework
- Standardisierte geheime AWS Manager-Schlüssel für mehrere Datenquellen
- Behebung von Leistungsregressionen
- Verbesserte Checkfunktion für PackedType
- Verbesserter Umgang mit LOW-VALUE für PackedType
- Verbesserte Spring Security-Verpackung für Cognito Connection
- CodeshiftPoint-Kodierung und -Dekodierung werden nicht auf Zieldatenbanken angewendet DB2

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 den Aufruf einer Prozedur in einer Prozedur
- Support für das Schlüsselwort OF im CICS-Befehl FORMATTIME bei mehrdeutigen Feldern
- JCL: Unterstützung für das Zeichen Â¥ in Variablen
- JCL: Berechnung von RC auf der Grundlage früherer Schritte
- Vergleichen 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 Hilfefunktion 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 LIKEREC
- 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 nun 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
- AS400
- 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 INSERT SQL TIME-Spalten im 24:00:00-Stundenformat (V7-10023) DB2
- 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 Lesetimeout 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)

AS400

Neue Features

- Implementierung der Hilfefunktion für AS4 00-Bildschirme (V7-9673)

### Verbesserungen

- Anzahl der Datensätze in INFDS (V7-9377)

### Transversale Fähigkeiten

### Neue Features

- Support für Runtime on EC2 zum Senden von Protokollen an Amazon CloudWatch (D87990246)
- Neuer Endpunkt zum Abrufen von Metriken über 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 (*INUx) (V7-7404)
- Verbesserte SQLDDS-Dateiumwandlung (V7-7687)
- Dateiobjekte werden nicht mehr für 00-Dateien generiert (V7-9062) AS4
- 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 COMP-Typ in AS4 00 (V7-9840)
- Das Analysieren von COBOL4 00 auf (Not) wurde verbessert (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)
- Im Zustand Else verbessert GOTOs (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 Migrationen von zOS und AS4 00. 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 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
- AS400
- Transversale Fähigkeiten

### zOS

#### Neue Features

- JCL - DynamicFileBuilder - Verbesserte Verwaltung von Datei-Handles (V7-9408)
- Verbesserte Formatkonvertierung einiger integrierter DB2 SQL-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

- Triggerung von gespeicherten Prozeduren für DB2 RDBMS hinzugefügt (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 handle different termination/dispositionin verketteten Datensätzen (V7-9653)
- pageSize für Blusam-Datensätze konfigurierbar machen (V7-9680)
- DSNUTIL — erlaubt das Laden von 24:00:00 als gültige ZEIT in LUW (V7-9697) DB2
- Support den Vergleich mit HOHEN WERTEN (0xff) in NumberUtils .ne ()/ NumberUtils.eq () (V7-9731)
- JCL/GROOVY - unterstützt DO... THEN-Schlüsselwörter in IF-THEN-ELSE IDCAMS-Klauseln zur Ausführung mehrerer Anweisungen in einem einzigen Block (V7-9750)
- Ungültiges JHDB hat das Programm außerhalb von Runner aufgerufen (V7-9782) JHDBBatch

- Support Leerzeichen auf der SORT OUTFIL-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 ()/() auf getNormalTermination (V7-10012) getAbnormalTermination ConcatenatedFileConfiguration

AS400

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
- AS400
- Transversale Fähigkeiten


zOS

Neue Features

- JCL - Verbesserte Protokollierung für das Ende des Vorgangs (V7-8509)
- PL1 - Verbesserte Beutelgenerierung 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 - Verbessern Sie die Unterstützung für GET EDIT mit Fixed-Point- und SYSIN-Stream (V7-9593)
- DB2 - Verbesserte Unterstützung für den Typ VARGRAPHIC (V7-9809) DB2
- CICS — Verbessert 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 CHARGRAPHIC als gültigen Parameter eines beliebigen Methodenaufrufs (V7-9589)

- Verbesserung der Auflösung von PL1 Hostvariablen bei Benennung mit bestimmten Zeichen @ # $ §. (V7-9654)
- COBOL - Support der Schlüsselwörter C01... C12 und S01... S05 als Parameter der WRITE ADVANCING-Anweisung beim Analyseschritt (V7-9669)

AS400

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 SQLModel und Generierung von AllIndex-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 Charset 0 threadsicher. CUSTOM93 (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 zOS- als auch für AS4 00-Legacy-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
- AS400
- 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)
- TSModel Das Abrufen erfolgt jetzt robust bei Bedarf (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)
- Integriert die Unterstützung von BUFFERSPACE/UNIQUE-Attributen in DEFINE-CLUSTER-Anweisungen (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 Endsperren 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)

AS400

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)
- Verbesserte Sortierreihenfolge und Abrufverwaltung zur Unterstützung des CL-Befehls OPNQRYF (Open Query File) und zusätzliche Unterstützung für den SHARE-Parameter in. 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)
- setOnGreaterOrEqual() ist nicht gleich 1 eingestellt (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. ResulSet (V7-9422)
- Verbesserung von JMS MQ — Unterstützung 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)
- Refaktorieren 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, Typ (V7-9542) Date/Time/Timestamp
- SQLCA aus der Analyse einbeziehen (V7-9333) FullFree
- 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 Â¬ durch! falls Â¬ 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 Sicherheitslücken in Blu Age

Common Vulnerabilities and Exposures (CVE) ist eine Referenzliste für öffentlich bekannte Cybersicherheitslücken. Jeder Eintrag enthält eine Identifikationsnummer, eine Beschreibung und mindestens einen öffentlichen Verweis.

Wir empfehlen, immer auf die neueste Version von AWS Blu Age zu aktualisieren, um vor bekannten Sicherheitslücken geschützt zu sein. Sicherheitsscans werden kontinuierlich mit Amazon Inspector durchgeführt und die Ergebnisse werden in NIST nach ihrem Schweregrad klassifiziert.

In der folgenden Liste werden die CVEs Fehler beschrieben, die in jeder verfügbaren Nebenversion behoben wurden und die sich aus der Verwendung von Abhängigkeiten ergeben:

| Version | CVE |
|---|---|
| 4.8.0 | CVE-2025-22228 |
| 4.7.0 | CVE-2025-25193, CVE-2025-24970, CVE-2024-47535, CVE-2024-38820, CVE-2024-38809, CVE-2024-22262, CVE-2022-45868 |
| 4.6.0 | CVE-2024-12801, CVE-2024-12798, CVE-2024-50379, CVE-2024-56337 |
| 4.5.0 | CVE-2024-47535, CVE-2024-52316, CVE-2024-47535, CVE-2024-38827 |
| 4.4.0 | CVE-2024-38820, CVE-2024-38821, CVE-2024-38809, CVE-2024-38816, |

| Version | CVE |
| --- | --- |
| | CVE-2024-47554, CVE-2024-6484, CVE-2024-6485 |
| 4.3.0 | CVE-2024-43788, CVE-2024-25898, CVE-2021-30246, CVE-2024-21484, CVE-2024-34750 |
| 4.2.0 | CVE-2020-11023, CVE-2023-26364, CVE-2019-11358, CVE-2020-11022, CVE-2021-23358, CVE-2017-18214, CVE-2023-24785, CVE-2023-48631 |
| 4.1.0 | CVE-2024-29025, CVE-2024-23080, CVE-2024-22262, CVE-2024-30171, CVE-2024-29857, CVE-2024-30172 |
| 4.0.0 | CVE-2016-1000027, CVE-2024-1471, CVE-2024-1597, CVE-2024-22243, CVE-2024-22233, CVE-2024-22234, CVE-2024-22259, CVE-2024-22257, CVE-2024-29131, CVE-2024-29133 |

> **Note**
>
> Für weitere Informationen zu Problemen in früheren Versionen CVEs wenden Sie sich bitte an Ihren Blu Age Delivery Manager AWS

## Upgrade-Anweisungen für AWS Blu Age

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

### Allgemeine Upgrades

In den meisten Fällen sollten Sie beim Upgrade der (nicht verwalteten) AWS Blu Age Runtime-Version die Artefakte (WARs, Konfigurationsdateien, Skripte usw.) Ihrer vorherigen Version durch die

in der neuen Version bereitgestellten ersetzen und Ihre Anwendung neu starten. Stellen Sie sicher, dass Sie nach dem Upgrade umfangreiche Regressionstests Ihrer modernisierten Anwendungen durchführen. Sie können sich auch an Ihren AWS Blu Age Delivery Manager wenden, um spezifische Anweisungen für Ihre Anwendung zu erhalten.

Informationen zum Upgrade der (verwalteten) Version von AWS Blu Age Runtime finden Sie unterVerwaltete Laufzeitumgebungen.

Bei einigen Upgrades ist möglicherweise eine zusätzliche Konfiguration erforderlich, um die Kompatibilität sicherzustellen. Folgen Sie in diesem Fall den Anweisungen für das jeweilige Upgrade.

## 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 log4 4j j-to-slf |

| Gruppe | ArtifactId | Änderung |
| --- | --- | --- |
| org.springframework.boot | spring-boot-starter-jta-Atomikos | Wechseln Sie zu com.atomikos: 3-starter:6.0.0 transactions-spring-boot |
| org.apache.commons | commons-dbcp2 | Führen Sie ein Upgrade auf 2.10.0 durch |
| org.postgresql | postgreql | 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 Lebenszyklus von Blu Age

In diesem Abschnitt werden die Enddaten (EOL) für die Hauptversionen von AWS Blu Age Runtime definiert. Auf diese Weise können Sie Versionsupgrades planen, sodass Sie immer über die neuesten Wartungsarbeiten und Funktionen auf dem Laufenden sind. Informationen zum Upgrade Ihrer Version finden Sie unterthe section called “ AWS Blu Age aktualisieren”.

Wir empfehlen, alle drei Monate nach neuen Versionen zu suchen und regelmäßig auf aktuelle Versionen zu aktualisieren. Bei jedem Upgrade sollten Sie Ihre modernisierten Anwendungen vor der Produktion oder kritischen Bereitstellungen ohne Regressionstests testen.

> **Note**
>
> Das Ende der Nutzungsdauer kann sich aufgrund kritischer Sicherheitsvorkehrungen ändern. Weitere Details finden Sie unter Lebenszyklus der Komponenten.

### AWS Ende der Lebensdauer (EOL) von Blu Age Runtime

In der folgenden Tabelle ist das EOL-Datum für jede Hauptversion zusammengefasst.

| Hauptversion | Datum des Lebensendes |
|---|---|
| Version 3 | 8. Juli 2024 |
| Version 4 | Muss noch veröffentlicht werden |

> **Note**
>
> Das EOL-Datum für die Hauptversion 4 wird an die Verfügbarkeit der nächsten Hauptversion angepasst.

Informationen zum Support-Modell für Nebenversionen finden Sie unterLebenszyklus der Komponenten.

# AWS Konzepte von Blu Age Runtime

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
- Was sind Datenvereinfacher in Blu Age AWS
- AWS Blu Age Blush
- Verfügbare Programme in der Utility-Webanwendung
- AWS Blu Age Blusam Verwaltungskonsole

## 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. Dazu dienen Bibliotheken, die Legacy-Konstrukte bereitstellen und die Organisation des Programmcodes standardisieren.

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 (Isofunktionalität), die Leistung (in Bezug auf die Ausführungszeit und den Ressourcenverbrauch) und die einfache Wartung modernisierter Programme durch Java-Entwickler, obwohl vertraute

Umgebungen und Redewendungen wie Tomcat, Spring, Getters/Setter, Fluent verwendet werden 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-Programmierumgebung 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, klassenbasiert, fließend). APIs Ein wichtiges Beispiel ist die Data Simplifier-Bibliothek, die ältere Speicherlayout- und Manipulationskonstrukte (wie sie in COBOL- oder RPG-Sprachen vorkommen) für Java-Programme bereitstellt. PL1 Diese JAR-Dateien sind eine Kernabhängigkeit des modernisierten Java-Codes, der aus älteren Programmen generiert wurde. Weitere Informationen zum Data Simplifier finden Sie unter. Was sind Datenvereinfacher in Blu Age AWS

### Webanwendung

Webanwendungsarchive (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 Runtime 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 von Anweisungen zur fließenden APIs Wiedergabe `EXEC CICS` von Java-Anweisungen,
  - 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 statelose Design ein wichtiger Mechanismus dafür. Daher basiert das Konzept der Staatenlosigkeit auf 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 (Redis OSS)
- 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.

Tatsächlich 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.

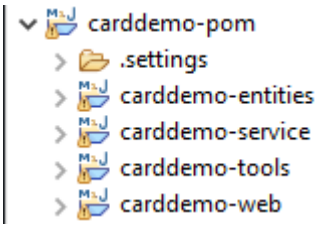
Themen

- 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.



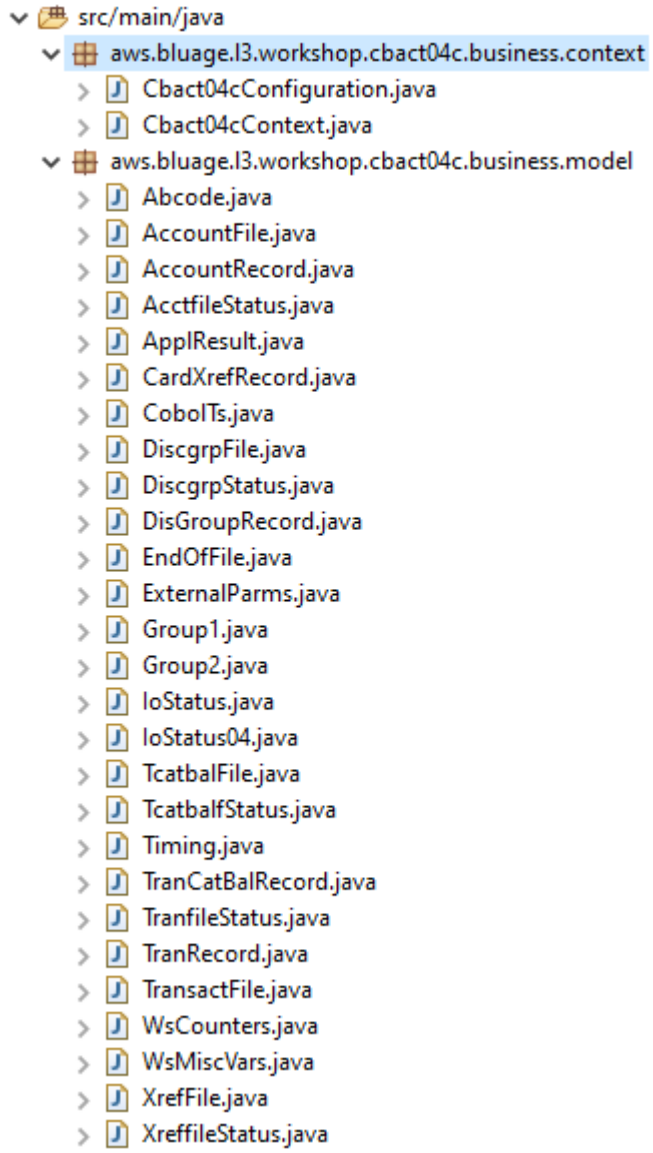

Die grundlegende Organisation basiert auf der folgenden Struktur:

- 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

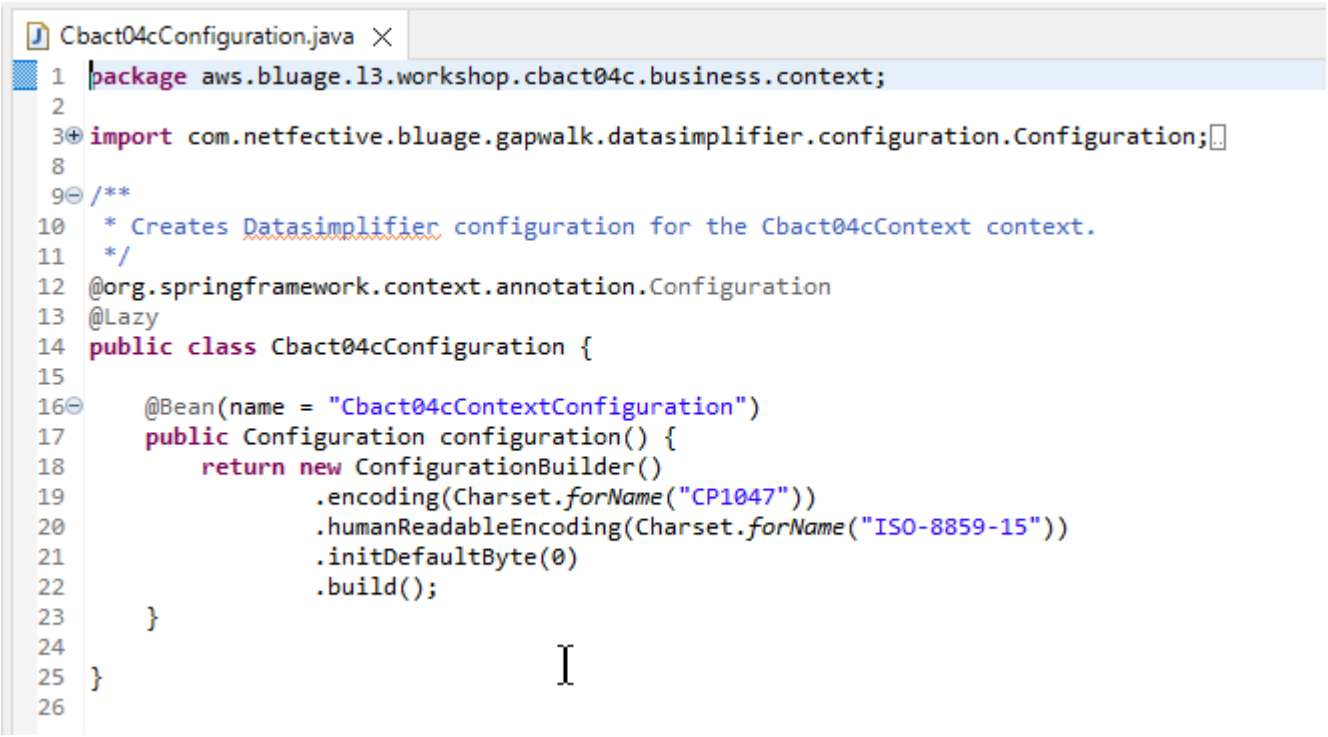
Jedes modernisierte Programm besteht aus zwei Paketen, einem business.context- und einem business.model-Paket.

- *`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.




- Eine Kontextklasse, die als Brücke zwischen den Programmdienstklassen (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.

Cbact04cContext.java

```
@Component("aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext")
@Import({
    aws.bluage.l3.workshop.cbact04c.business.model.TcatbalFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.XrefFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.AccountFile.class
,   aws.bluage.l3.workshop.cbact04c.business.model.TransactFile.class
})
@Lazy
@Scope("prototype")
public class Cbact04cContext extends JicsRuntimeContext {

    @Autowired
    private TcatbalFile tcatbalFile;

    @Autowired
    private XrefFile xrefFile;

    @Autowired
    private DiscgrpFile discgrpFile;

    @Autowired
    private AccountFile accountFile;

    @Autowired
    private TransactFile transactFile;

    private IndexedFile tcatbalFileFile;

    private IndexedFile xrefFileFile;

    private IndexedFile discgrpFileFile;

    private IndexedFile accountFileFile;

    private SequentialFile transactFileFile;

    private TranCatBalRecord tranCatBalRecord;
    private TcatbalfStatus tcatbalfStatus;
    private CardXrefRecord cardXrefRecord;
```

- *base package*.*program*.business.model

  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. Was sind Datenvereinfacher in Blu Age AWS

DiscgrpFile.java

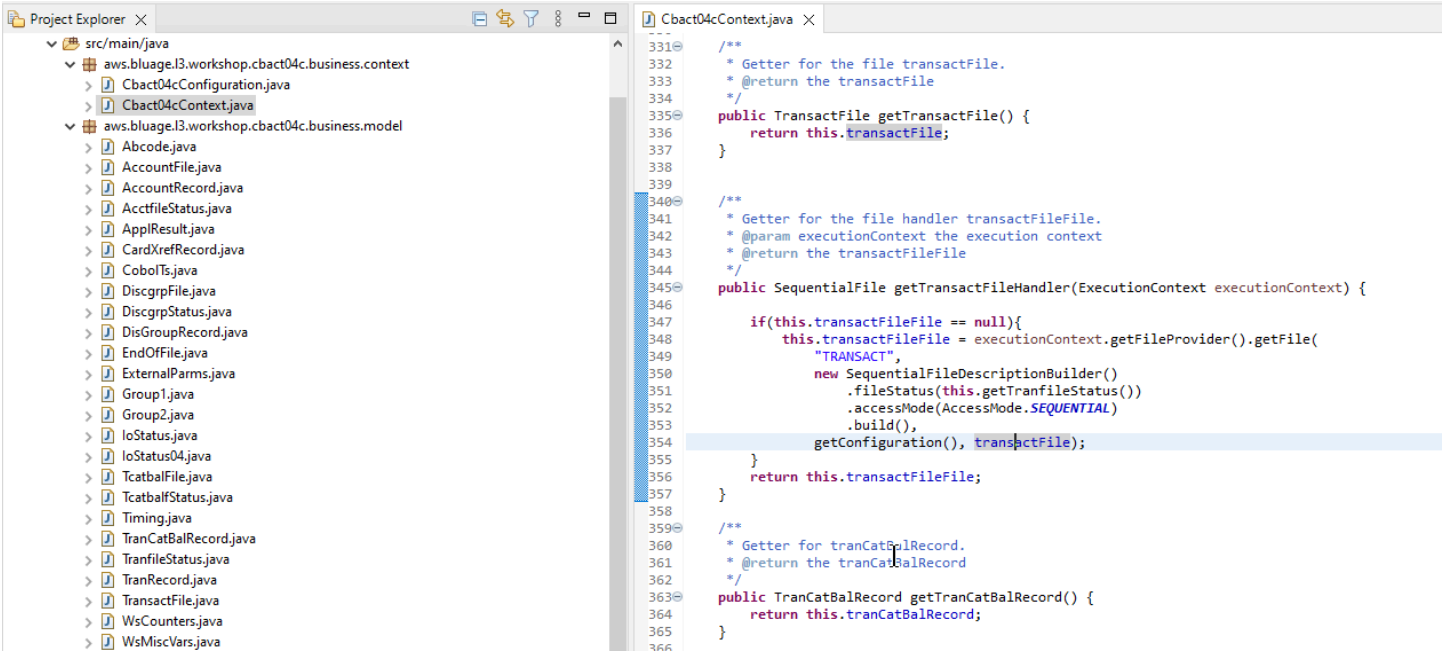
```java
package aws.bluage.l3.workshop.cbact04c.business.model;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary;
import com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group;
import com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference;
import com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference;
import com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity;
import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType;
import com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType;
import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Lazy;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

/**
 * Data simplifier file DiscgrpFile.
 *
 * <p>About 'fdDiscgrpRec' field, <br>uml entity: aws.bluage.l3.workshop.cbact04c.business.model.FdDiscgrpRec
 * <br></p>
 *
 */
@Component("aws.bluage.l3.workshop.cbact04c.business.model.DiscgrpFile")
@Lazy
@Scope("prototype")
public class DiscgrpFile extends RecordEntity {

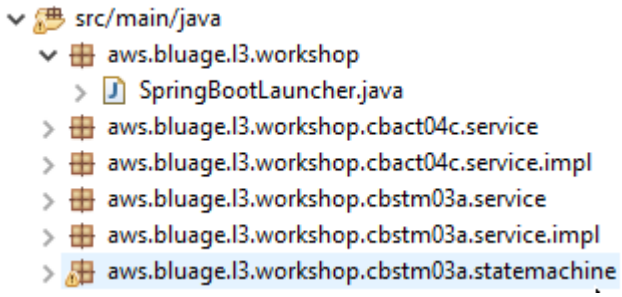
    private final Group root = new Group(getData());
    private final Group fdDiscgrpRec = new Group(root);
    private final Group fdDiscgrpKey = new Group(fdDiscgrpRec);
    private final Elementary fdDisAcctGroupId = new Elementary(fdDiscgrpKey,new AlphanumericType(10));
    private final Elementary fdDisTranTypeCd = new Elementary(fdDiscgrpKey,new AlphanumericType(2));
    private final Elementary fdDisTranCatCd = new Elementary(fdDiscgrpKey,new ZonedType(4, 0, false));
    private final Elementary fdDiscgrpData = new Elementary(fdDiscgrpRec,new AlphanumericType(34));
```

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:



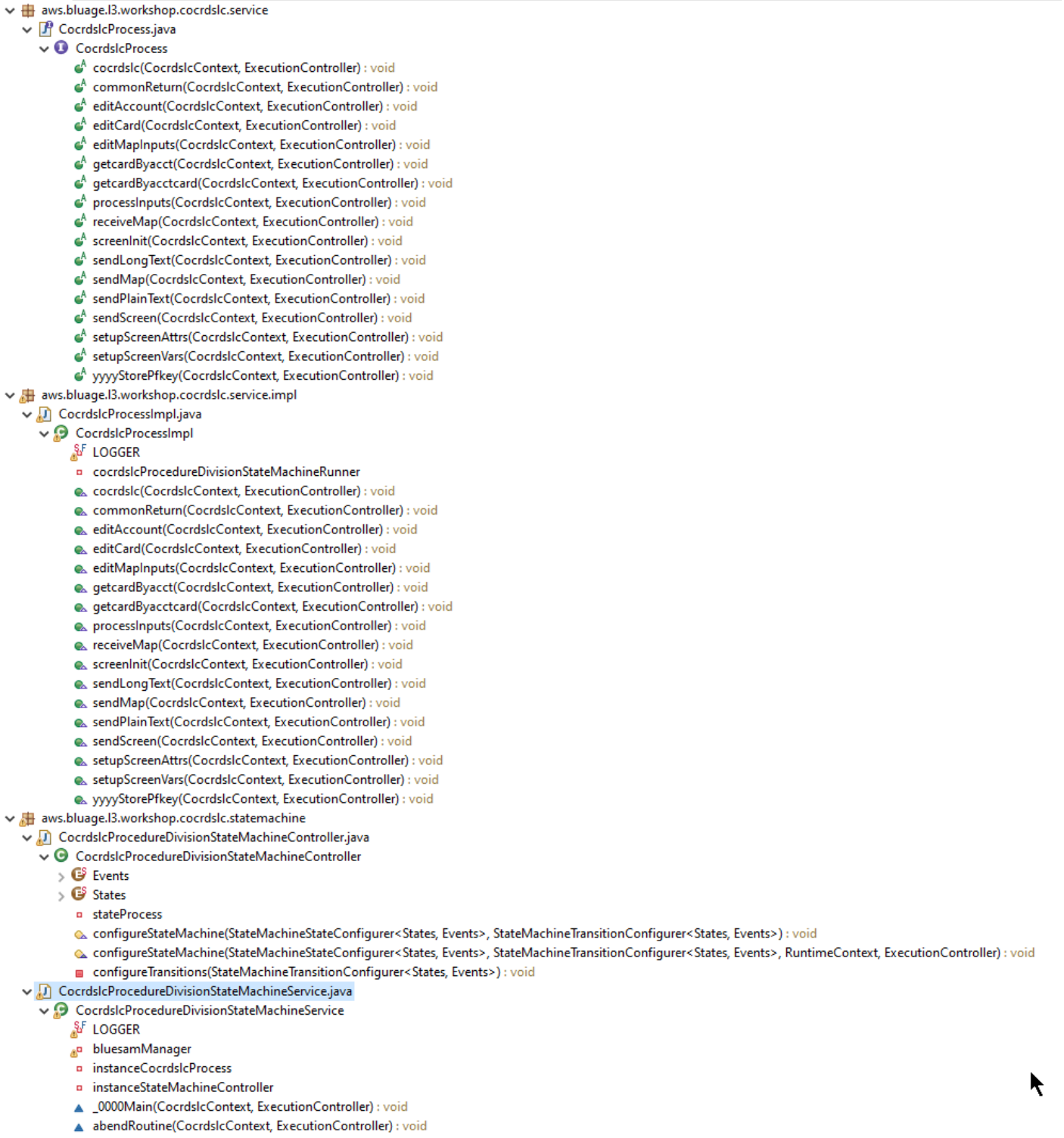

Diese Klasse ist insbesondere verantwortlich für:

- 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:




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 des alten Kontrollflusses ein State-Machine-Ansatz (d. h. die Verwendung des StateMachine Spring-Frameworks) verwendet werden muss, um den alten Ausführungsablauf ordnungsgemäß abzudecken.

In diesem Fall enthält das Statemachine-Unterpaket zwei Klassen:

- *Program*`ProcedureDivisionStateMachineController`: eine Klasse, die eine Klasse erweitert, die die Schnittstellen `StateMachineController` (definiert Operationen, die zur Steuerung der Ausführung einer Zustandsmaschine benötigt werden) 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.

CocrdslcProcedureDivisionStateMachineController.java

```
package aws.bluage.l3.workshop.cocrdslc.statemachine;

import aws.bluage.l3.workshop.cocrdslc.business.context.CocrdslcContext;

/**
 * Controller managing the state machine "CocrdslcProcedureDivisionStateMachine" execution.
 */
@Component("aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineController")
@Import({
aws.bluage.l3.workshop.cocrdslc.statemachine.CocrdslcProcedureDivisionStateMachineService.class
})
@Lazy
public class CocrdslcProcedureDivisionStateMachineController extends SimpleStateMachineController<States, Events> {

    /**
     * State machine states.
     */
    public enum States {
        _0000_MAIN_1, _0000_MAIN, ABEND_ROUTINE, FINAL, LOCAL_FINAL
    }

    /**
     * State machine events.
     */
    public enum Events {
        TO__0000_MAIN_1, TO__0000_MAIN, TO_ABEND_ROUTINE, TO_FINAL, TO_LOCAL_FINAL
    }

    /**
     * State machine state process service provider.
     */
    @Autowired
    @Lazy
    private CocrdslcProcedureDivisionStateMachineService stateProcess;

    @Override
    protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
        throw new UnsupportedOperationException("Please use the four arguments configureStateMachine method instead: configureStateMachine(StateMachineStateConfigurer<States, Events> states, "
                + "StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl)");
    }

    @Override
    protected void configureStateMachine(StateMachineStateConfigurer<States, Events> states, StateMachineTransitionConfigurer<States, Events> transitions, RuntimeContext ctx, ExecutionController ctrl) throws Exception {
        StateConfigurer<States, Events> configurer = states.withStates();
        configurer.initial(States._0000_MAIN_1).end(States.FINAL);
        configurer.state(States._0000_MAIN_1);
        configurer.state(States.FINAL);

        StateConfigurer<States, Events> subConfigurer = states.withStates().parent(States._0000_MAIN_1);
        subConfigurer.initial(States._0000_MAIN).end(States.LOCAL_FINAL);
        CocrdslcContext lctx = (CocrdslcContext) ctx;
        subConfigurer.state(States._0000_MAIN, buildAction(() -> {stateProcess._0000Main(lctx, ctrl);}), null);
        subConfigurer.state(States.ABEND_ROUTINE, buildAction(() -> {stateProcess.abendRoutine(lctx, ctrl);}), null);

        configureTransitions(transitions);
    }

    /**
     * Declare state machine transitions.
     * @param transitions the transitions configuration helper
     */
    private void configureTransitions(StateMachineTransitionConfigurer<States, Events> transitions) throws Exception {
        transitions.withLocal().source(States._0000_MAIN_1).target(States.ABEND_ROUTINE).event(Events.TO_ABEND_ROUTINE);
        transitions.withExternal().source(States.ABEND_ROUTINE).target(States.FINAL).event(Events.TO_FINAL);
    }
}

```

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);
```

- *Program*ProcedureDivisionStateMachineService: 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:     COCRDSLC.CBL                                      *
72        Layer:       Business logic                                    *
73        Function:    Accept and process credit card detail request     *
74        *****************************************************************
75        Copyright Amazon.com, Inc. or its affiliates.
76        All Rights Reserved.
77        Licensed under the Apache License, Version 2.0 (the \"License\").
78        You may not use this file except in compliance with the License.
79        You may obtain a copy of the License at
80        http://www.apache.org/licenses/LICENSE-2.0
81        Unless required by applicable law or agreed to in writing,
82        software distributed under the License is distributed on an
83        \"AS IS\" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND,
84        either express or implied. See the License for the specific
85        language governing permissions and limitations under the License
86        *****************************************************************
87        Ver: CardDemo_v1.0-15-g27d6c6f-68 Date: 2022-07-19 23:16:00 CDT */
88        instanceStateMachineController.registerSignalHandler(Events.TO_ABEND_ROUTINE, "!ABEND");
89        HandleAbendBuilder.newInstance(ctx.getDfheiblk(), ctx).execute().handleException();
90        ctx.getCcWorkAreas().getCcWorkAreaReference().getField().initialize();
91        ctx.getWsMiscStorage().getField().initialize();
92        DataUtils.initialize(ctx.getWsCommarea().getWsCommareaReference());
93
```

CocrdslcProcedureDivisionStateMachineService.java

```
221      *
222      * @param ctx
223      * @param ctrl
224      */
225     void abendRoutine(CocrdslcContext ctx, ExecutionController ctrl) {
226         if (DataUtils.isLowValue(ctx.getAbendData().getAbendMsgReference())) {
227             ctx.getAbendData().setAbendMsg("UNEXPECTED ABEND OCCURRED.");
228         }
229         ctx.getAbendData().setAbendCulprit(ctx.getWsLiterals().getLitThispgm());
230         SendTextBuilder.newInstance(ctx.getDfheiblk(), ctx)
231         .withData(ctx.getAbendData())
232         .withLength(134)
233         .execute();
234         HandleAbendBuilder.newInstance(ctx.getDfheiblk(), ctx).cancel().execute().handleException();
235         AbendBuilder.newInstance(ctx.getDfheiblk(), ctx).withAbendCode("9999").execute().handleException();
236
237         /*
238         Ver: CardDemo_v1.0-15-g27d6c6f-68 Date: 2022-07-19 23:12:33 CDT */
239         instanceStateMachineController.sendEvent(Events.TO_FINAL);
240
241     }
242 }
243
```

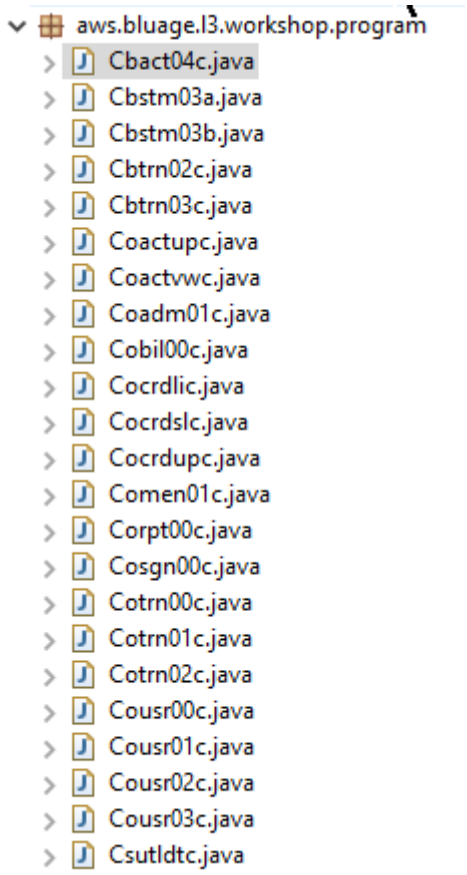
Der Statemachine-Dienst ruft auch die zuvor beschriebene Implementierung des Prozessdienstes auf:

CocrdslcProcedureDivisionStateMachineService.java

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

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

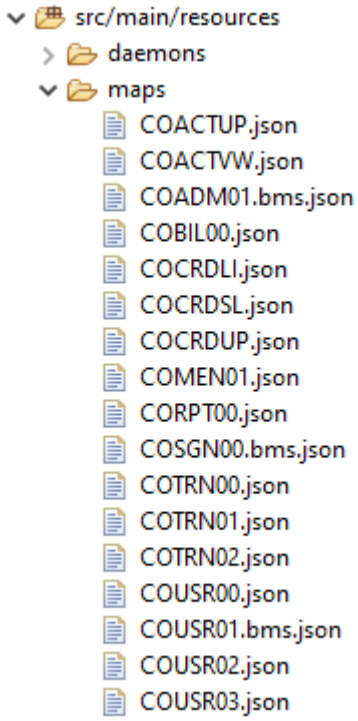
(mehr dazu später). Jede Klasse implementiert die `Program` Schnittstelle, die Markierung für einen Programmeinstiegspunkt.




Andere Artefakte

- BMS-Begleiter MAPs

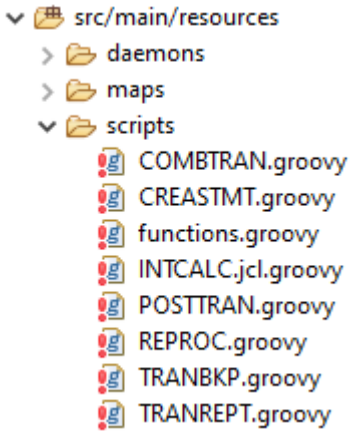
  Neben programmbezogenen Artefakten kann das Serviceprojekt auch andere Artefakte für verschiedene Zwecke enthalten. Bei der Modernisierung einer CICS-Online-Anwendung wird im Rahmen des Modernisierungsprozesses eine JSON-Datei erstellt und im the /src/main/resources Kartenordner des Ordners abgelegt:




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

- Groovy-Skripte

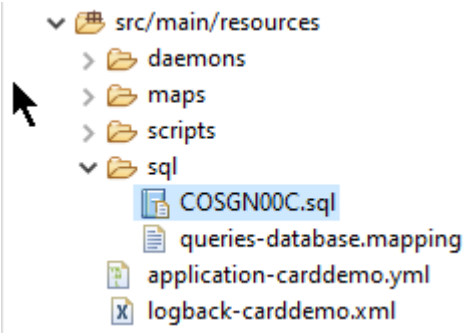
  Wenn die alte Anwendung über JCL-Skripte verfügte, wurden diese als Groovy-Skripte modernisiert und in einem the /src/main/resources/scripts Ordner gespeichert (mehr zu diesem speziellen Ort später):



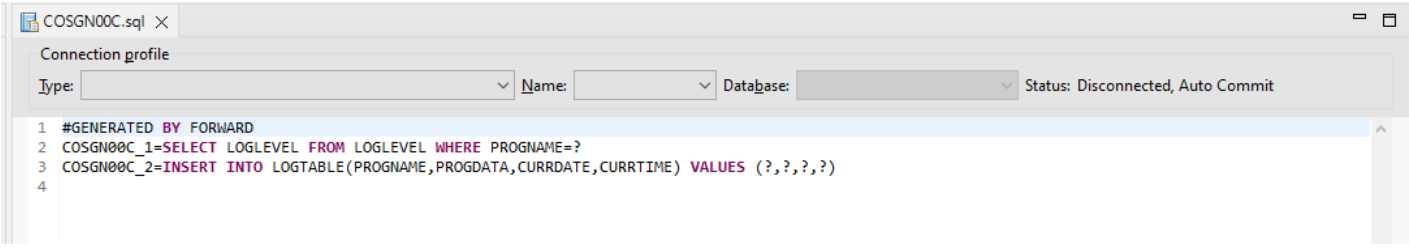

  Diese Skripts werden zum Starten von Batch-Jobs (dedizierte, nicht interaktive, rechenintensive Datenverarbeitungs-Workloads) verwendet.

- SQL-Dateien

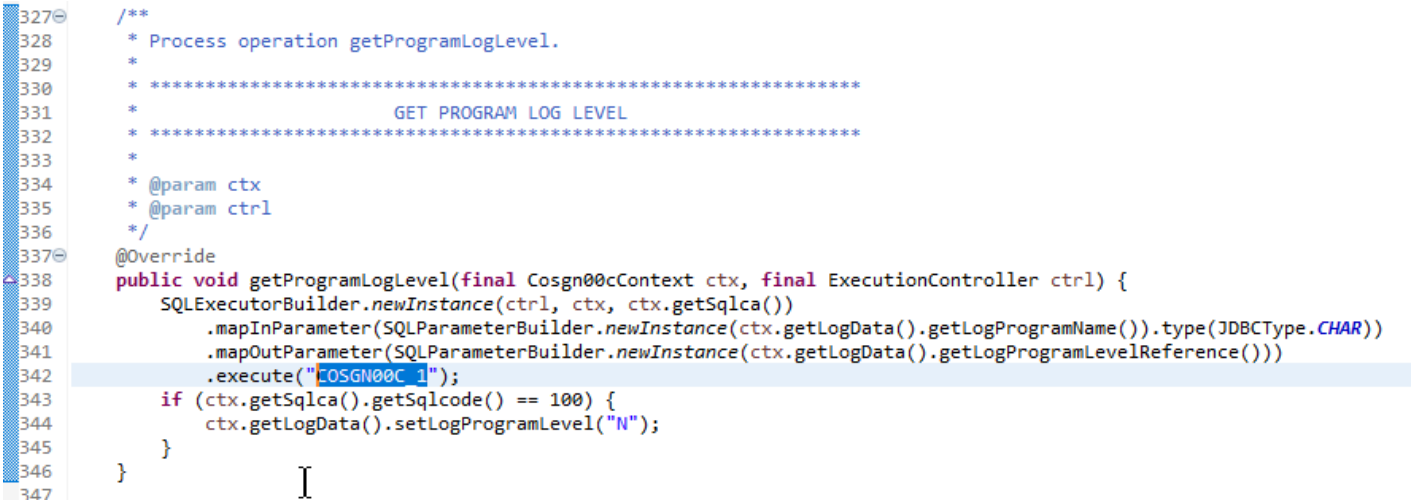
Wenn die alte Anwendung SQL-Abfragen verwendete, wurden die entsprechenden modernisierten SQL-Abfragen in speziellen Eigenschaftendateien mit dem Benennungsmuster Programm .sql gesammelt, wobei Programm der Name des Programms ist, das diese Abfragen verwendet.




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

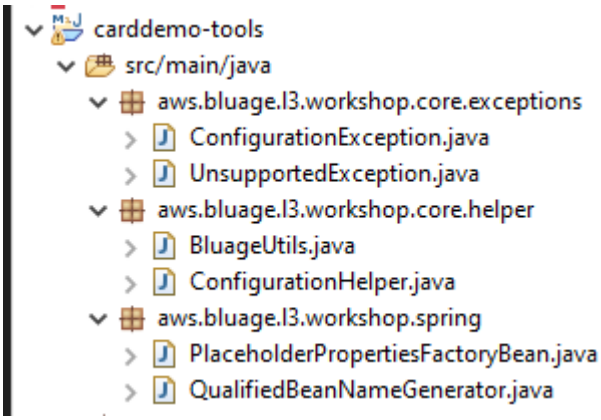



Zum Beispiel führt das Programm COSGN00C die Abfrage mit dem Schlüssel „COSGN00C_1" (dem ersten Eintrag in der SQL-Datei) aus:



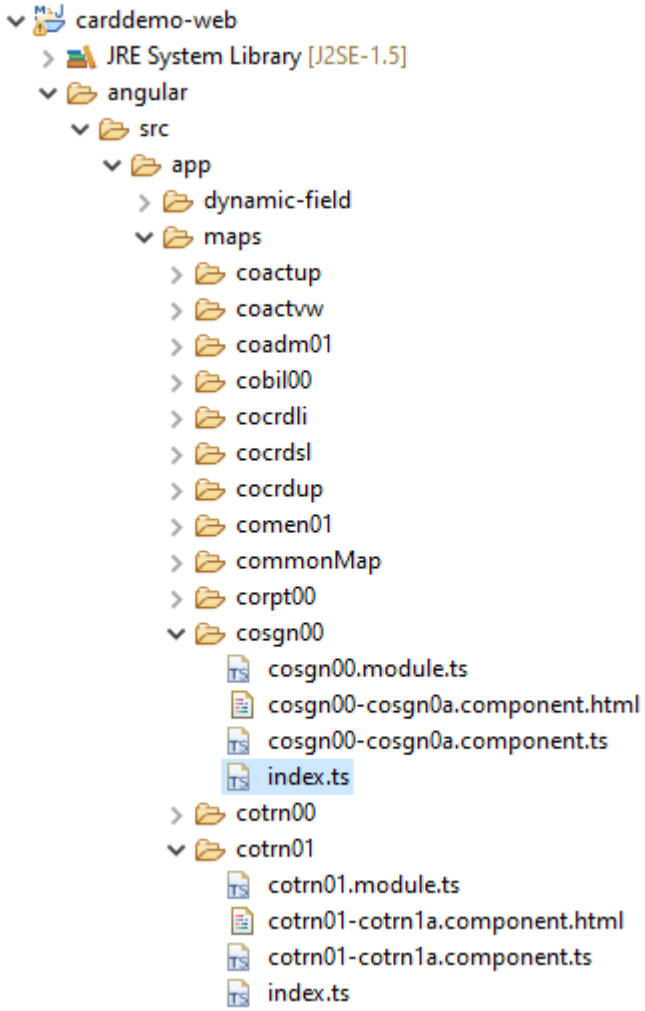
### Inhalt des Utilities-Projekts

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 dient zur Darstellung der Benutzeroberflächenbildschirme MAPs . 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 `Program` Schnittstelle 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

```java
package aws.bluage.l3.workshop.program;

import aws.bluage.l3.workshop.SpringBootLauncher;

/**
 * Reference the spring application of program CBACT04C.
 *  Provides an access to the contained program for the run unit.
 */
@Component
@Import({
    aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cConfiguration.class,
    aws.bluage.l3.workshop.cbact04c.business.context.Cbact04cContext.class,
    aws.bluage.l3.workshop.cbact04c.service.impl.Cbact04cProcessImpl.class
})
public class Cbact04c implements Program {
    /**
     * Unique identifiers for the contained program.
     */
    private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("CBACT04C").collect(Collectors.toSet()));

    /**
     * Main program identifier for the contained program.
     */
    private static final String programIdentifier = "CBACT04C";
    @Autowired
    PlatformTransactionManager transactionManager;

    @Autowired
    Map<String, DataSource> datasources;
    @Autowired
    BeanFactory beanFactory;
    /**
     * {@inheritDoc}
     */
    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        return SpringBootLauncher.getCac();
    }

    /**
     * {@inheritDoc}
     */
    @Override
    public void updateExecutionContext(ExecutionContext executionContext) {
        executionContext.setDatasources(datasources);
        executionContext.setDatabaseSupport(ExecutionContext.DatabaseSupport.POSTGRE);
        executionContext.setSqlcaVersion(ExecutionContext.SqlcaVersion.getEnum("ansi-comp5"));
        executionContext.setTransactionManager(transactionManager);
        executionContext.setUseSQLDateNewParadigm(true);
        executionContext.setUseSQLTrimStringType(false);
    }

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

In 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. Die Gapwalk-Utility-Pgm Webapp stellt beispielsweise 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 wurden, 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 befinden. the /src/main/resources/scripts Die Ordnerhierarchie für Skripts wird durchquert, und alle gefundenen Groovy-Skripten (mit Ausnahme des reservierten Skripts special functions.groovy) werden in der registriert`ScriptRegistry`, wobei ihr Kurzname (der Teil des Skriptdateinamens, der vor dem ersten Punktzeichen steht) 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 beim Registrierungsprozess nicht berücksichtigt: .groovy. typically /scripts/A/myscript.groovy and /scripts/B/myscript.groovy will lead to /scripts/B/myscript.groovy overwriting /scripts/A/myscript

Die Groovy-Skripte im the /src/main/resources/daemons Ordner 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 a 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

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

- 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 ist`CallBuilder`, 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 an:

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 Parameter (ein `Record` 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 Anrufer 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`
- byValueWithGrenzen (Object, int, int): Fügt ein Argument hinzu, das in ein umgewandelt wird, und extrahiert den Teil des Byte-Arrays`Record`, 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

```groovy
// Import
import com.netfective.bluage.gapwalk.rt.provider.ScriptRegistry
import com.netfective.bluage.gapwalk.rt.call.MainProgramRunner
import com.netfective.bluage.gapwalk.io.support.FileConfigurationUtils
import com.netfective.bluage.gapwalk.rt.job.support.DefaultJobContext
import com.netfective.bluage.gapwalk.rt.utils.GroovyUtils
import com.netfective.bluage.gapwalk.rt.io.support.FileConfiguration
import com.netfective.bluage.gapwalk.rt.shared.AbendException
import com.netfective.bluage.gapwalk.rt.call.exception.GroovyExecutionException
// Variables
mpr = applicationContext.getBean("com.netfective.bluage.gapwalk.rt.call.ExecutionController", MainProgramRunner.class)
TreeMap mapTransfo = [:]
```

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

```
//*******************************************************************
//*                              STEPS                              *
//*******************************************************************
// STEP PRC001 - PGM - IDCAMS*****************************************
def stepPRC001(Object shell, Map params, Map programResults){
    shell.with {
        if (checkValidProgramResults(programResults)) {
            return execStep("PRC001", "IDCAMS", programResults, {
                mpr
                    .withFileConfigurations(new FileConfigurationUtils()
                        .systemOut("SYSPRINT")
                        .output("*")
                        .build()
                        .bluesam("FILEIN")
                        .dataset("NULLFILE")
                        .disposition("SHR")
                        .build()
                        .bluesam("FILEOUT")
                        .dataset("NULLFILE")
                        .disposition("SHR")
                        .build()
                        .fileSystem("SYSIN")
                        .path("&CNTLLIB(REPROCT)")
                        .disposition("SHR")
                        .build()
                        .getFileConfigurations(fcmap))
                    .withParameters(params)
                    .runProgram("IDCAMS")
                })
        }
    }
}
```

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.

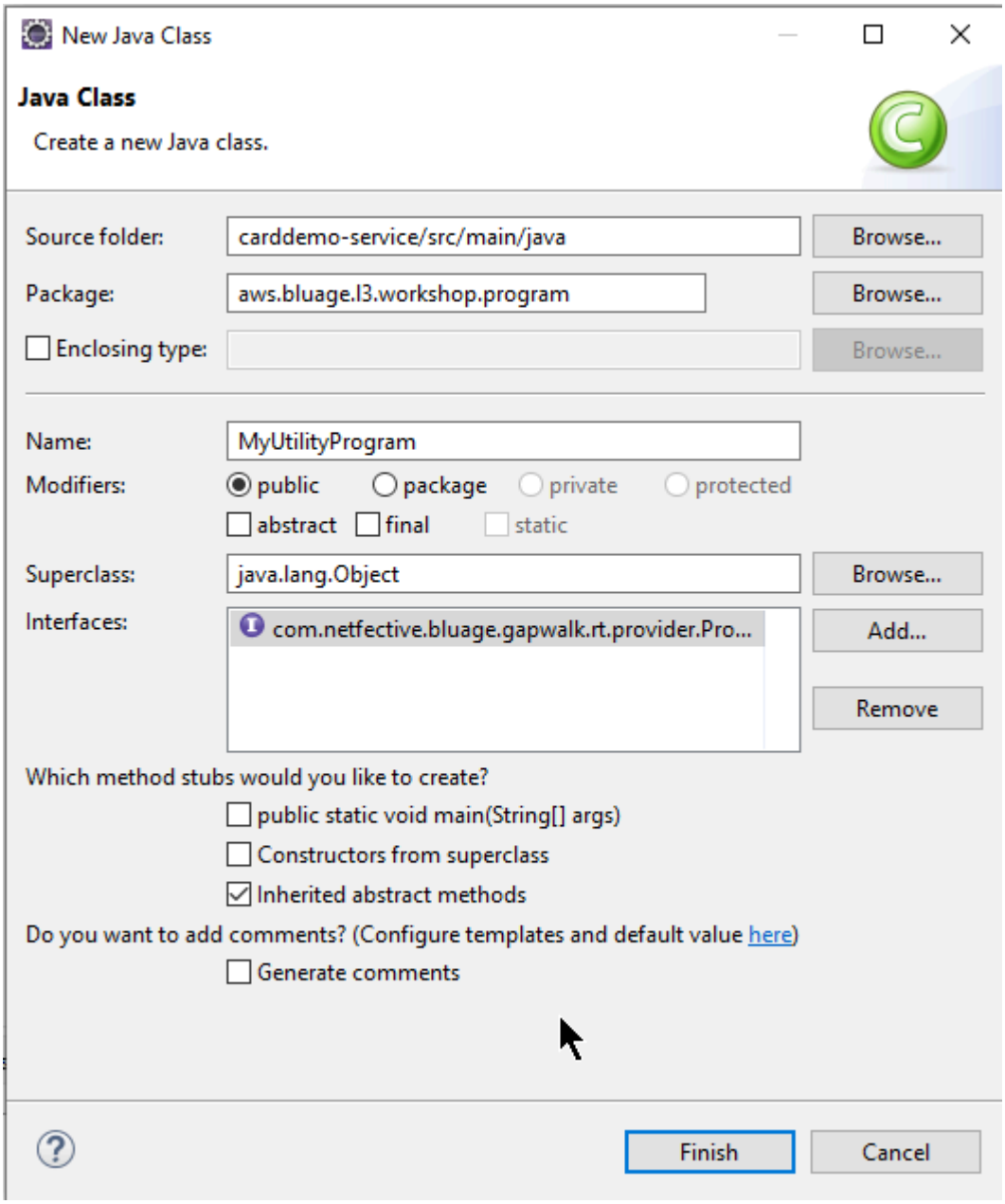
In diesem Fall müssen Sie das fehlende Programm möglicherweise selbst in Java schreiben (vorausgesetzt, Sie haben genügend Wissen über das erwartete Verhalten des Programms, das Speicherlayout der Programmargumente, falls vorhanden, usw.). Ihr Java-Programm muss den in diesem Dokument beschriebenen Programmmechaniken 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

```java
package aws.bluage.l3.workshop.program;

import java.util.Set;

public class MyUtilityProgram implements Program {

    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public Set<String> getProgramIdentifiers() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public Context getContext() {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void run(ExecutionController ctrl) {
        // TODO Auto-generated method stub

    }

}
```

### Frühling-Integration

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

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

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

import aws.bluage.l3.workshop.SpringBootLauncher;

@Component
public class MyUtilityProgram implements Program {
```

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() {...}

    @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.

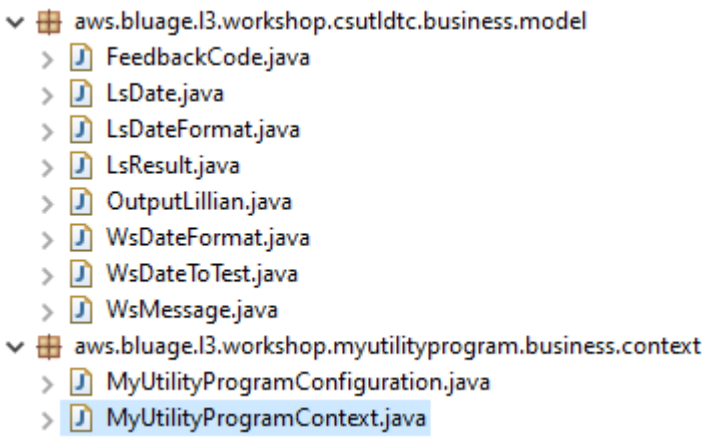
Unter Kurse im Zusammenhang mit dem Programm Bezugnahme auf können Sie sehen, dass ein Programm mindestens zwei Begleitklassen benötigt:

- 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:




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

```java
package aws.bluage.l3.workshop.myutilityprogram.business.context;

import java.nio.charset.Charset;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Lazy;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.datasimplifier.configuration.ConfigurationBuilder;

/**
 * Creates Datasimplifier configuration for the MyUtilityProgram context.
 */
@org.springframework.context.annotation.Configuration
@Lazy
public class MyUtilityProgramConfiguration {

    @Bean(name = "MyUtilityProgramContextConfiguration")
    public Configuration configuration() {
        return new ConfigurationBuilder()
                .encoding(Charset.forName("CP1047"))
                .humanReadableEncoding(Charset.forName("ISO-8859-15"))
                .initDefaultByte(0)
                .build();
    }
}

```

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

```java

import org.springframework.beans.factory.annotation.Qualifier;
import org.springframework.context.annotation.Lazy;
import org.springframework.context.annotation.Scope;
import org.springframework.stereotype.Component;

import com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration;
import com.netfective.bluage.gapwalk.rt.context.RuntimeContext;

@Component("aws.bluage.l3.workshop.mytuilityprogram.business.context.MyUtilityProgramContext")
@Lazy
@Scope("prototype")
public class MyUtilityProgramContext extends RuntimeContext{

    protected MyUtilityProgramContext(@Qualifier("MyUtilityProgramContextConfiguration") Configuration configuration) {
        super(configuration);
    }

    @Override
    public void cleanUp() {
        // TODO implement clean-up of associated data structures if any
    }

    @Override
    protected void doReset() {
        // TODO implement reset of associated data structures if any
    }

}

```

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 abzuzielen.
- 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

```java
import aws.bluage.l3.workshop.SpringBootLauncher;

@Component
@Import({
    aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramConfiguration.class,
    aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class
})
public class MyUtilityProgram implements Program {

    @Autowired
    BeanFactory beanFactory;

    /**
     * Unique identifiers for the contained program.
     */
    private static final Set<String> programIdentifiers = Collections.unmodifiableSet(Stream.of("MYUTILPG").collect(Collectors.toSet()));

    private static final String programIdentifier = "MYUTILPG";

    @Override
    public ConfigurableApplicationContext getSpringApplication() {
        return SpringBootLauncher.getCac();
    }

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

    /**
     * {@inheritDoc}
     */
    @Override
    public String getProgramMainIdentifier() {
        return programIdentifier;
    }


    @Override
    public Context getContext() {
        return ProgramContextStore.getOrCreate(
        getProgramMainIdentifier(),
        aws.bluage.l3.workshop.myutilityprogram.business.context.MyUtilityProgramContext.class,
        beanFactory);
    }

```

Hinweise:

- Die getContext-Methode muss strikt wie gezeigt implementiert werden, wobei eine Delegierung 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 speichern`ProgramContextStore`; 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. Anregungen dazu finden Sie in der Initialisierungsmethode der generierten SpringbootLauncher Klassen in den Dienstprojekten (sieheInhalt 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 |
| --- | --- |
| `CallBuilder` | `com.netfective.bluage.gapwalk.runtime.statements.CallBuilder` |
| `Configuration` | `com.netfective.bluage.gapwalk.datasimplifier.configuration.Configuration` |

| 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.rt.statemachine.SimpleStateMachineController |
| StateMachineController | com.netfective.bluage.gapwalk.rt.statemachine.StateMachineController |
| StateMachineRunner | com.netfective.bluage.gapwalk.rt.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 |

# Was sind Datenvereinfacher in Blu Age AWS

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-Programmierumgebung 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 RPG-Datenlayouts als auch PL1 für die Modernisierung des RPG-Datenlayouts verwenden, da die meisten Konzepte ähnlich sind.

Themen


- Die wichtigsten Klassen
- Datenbindung und Zugriff
- Häufig gestellte Fragen zu den besprochenen Java-Typen

## Die wichtigsten 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.

`Group`Felder 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 + (size in bytes of first field)` usw. Sie werden verwendet, um Sequenzen von COBOL-Feldern unter demselben enthaltenden Feld darzustellen.

`Union`Felder 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. `index * (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 Bytegröße`Record`, um die Instanziierung der ihnen zugrunde liegenden Klassen. `RecordEntity`

In anderen COBOL-Fällen, wie z. B. beim Zuordnen von LINKAGE-Elementen zu Programmargumenten oder bei der Modernisierung des SET ADDRESS OF-Konstrukts, muss eine Instanz mit einem bereitgestellten Objekt verknüpft 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` Implementationen `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 |
|---|---|
| `Alphanumeric` | `com.netfective.bluage.gapwalk.datasimplifier.elementary.Alphanumeric` |
| `AlphanumericType` | `com.netfective.bluage.gapwalk.datasimplifier.metadata.type.AlphanumericType` |
| `BinaryType` | `com.netfective.bluage.gapwalk.datasimplifier.metadata.type.BinaryType` |
| `Bindable` | `com.netfective.bluage.gapwalk.datasimplifier.data.Bindable` |

| Kurzer Name | Vollständig qualifizierter Name |
| --- | --- |
| Elementary | com.netfective.bluage.gapwalk.datasimplifier.data.structure.Elementary |
| ElementaryRangeReference | com.netfective.bluage.gapwalk.datasimplifier.entity.ElementaryRangeReference |
| Filler | com.netfective.bluage.gapwalk.datasimplifier.data.structure.Filler |
| Group | com.netfective.bluage.gapwalk.datasimplifier.data.structure.Group |
| PackedType | com.netfective.bluage.gapwalk.datasimplifier.metadata.type.PackedType |
| Primitive | com.netfective.bluage.gapwalk.datasimplifier.elementary.Primitive |
| RangeReference | com.netfective.bluage.gapwalk.datasimplifier.entity.RangeReference |
| RangeType | com.netfective.bluage.gapwalk.datasimplifier.metadata.type.RangeType |
| Record | com.netfective.bluage.gapwalk.datasimplifier.data.Record |

| Kurzer Name | Vollständig qualifizierter Name |
| --- | --- |
| `RecordAdaptable` | `com.netfective.bluage.gapwalk.datasimplifier.data.RecordAdaptable` |
| `RecordEntity` | `com.netfective.bluage.gapwalk.datasimplifier.entity.RecordEntity` |
| `Repetition` | `com.netfective.bluage.gapwalk.datasimplifier.data.structure.Repetition` |
| `Union` | `com.netfective.bluage.gapwalk.datasimplifier.data.structure.Union` |
| `ZonedDecimal` | `com.netfective.bluage.gapwalk.datasimplifier.elementary.ZonedDecimal` |
| `ZonedType` | `com.netfective.bluage.gapwalk.datasimplifier.metadata.type.ZonedType` |

## AWS Blu Age Blush

Auf Mainframe-Systemen (im Folgenden als „Legacy“ bezeichnet) werden Geschäftsdaten häufig mithilfe von VSAM (Virtual Storage Access Method) gespeichert. Die Daten werden in „Datensätzen“ (Byte-Arrays) gespeichert, die zu einem „Datensatz“ gehören.

Es gibt vier Datensatzorganisationen:

- KSDS: Datensätze mit Schlüsselsequenz — Datensätze werden anhand eines Primärschlüssels (keine doppelten Schlüssel zulässig) und optional durch zusätzliche „alternative“ Schlüssel indexiert. Alle Schlüsselwerte sind Teilmengen des Datensatz-Byte-Arrays, wobei jeder Schlüssel wie folgt definiert ist:

  - ein Offset (0-basiert, 0 ist der Anfang des Inhalts des Datensatz-Byte-Arrays, gemessen in Byte)
  - eine Länge (ausgedrückt in Byte)
  - ob es doppelte Werte toleriert oder nicht
- ESDS: Datensätze mit Eingangssequenz — auf Datensätze wird meistens sequentiell zugegriffen (basierend auf ihrer Einfügungsreihenfolge im Datensatz), es können jedoch zusätzliche alternative Schlüssel verwendet werden;
- RRDS: Relative Datensätze — der Zugriff auf Datensätze erfolgt über „Sprünge“, d. h. unter Verwendung relativer Datensatznummern; die Sprünge können entweder vorwärts oder rückwärts erfolgen;
- LDS: Lineare Datensätze — da gibt es keine Datensätze, sondern lediglich einen Bytestrom, der in Seiten organisiert ist. Wird hauptsächlich für interne Zwecke auf älteren Plattformen verwendet.

Bei der Modernisierung älterer Anwendungen mithilfe des AWS Blu Age-Refactoring-Ansatzes sind modernisierte Anwendungen nicht mehr für den Zugriff auf VSAM-gespeicherte Daten vorgesehen, wobei die Datenzugriffslogik beibehalten wird. Die Blusam-Komponente ist die Antwort: Sie ermöglicht den Import von Daten aus veralteten VSAM-Datensätzen, bietet eine API für die modernisierte Anwendung, um sie zu bearbeiten, sowie eine spezielle Verwaltungs-Webanwendung. Siehe the section called “Blusam Verwaltungskonsole”.

> **Note**
>
> Blusam unterstützt nur KSDS, ESDS und RRDS.

Die Blusam-API ermöglicht die Beibehaltung der Datenzugriffslogik (sequentielle, zufällige und relative Lesevorgänge; Datensätze einfügen, aktualisieren und löschen), wohingegen die Komponentenarchitektur, die auf einer Mischung aus Caching-Strategien und RDBMS-basiertem Speicher basiert, I/O-Operationen mit hohem Durchsatz und begrenzten Ressourcen ermöglicht.

## Blusam-Infrastruktur

Blusam verwendet PostgreSQL RDBMS für die Speicherung von Datensätzen, sowohl für Rohdaten als auch für Schlüsselindizes (falls zutreffend). Die bevorzugte Option ist die Verwendung der Amazon Aurora PostgreSQL-kompatiblen Engine. Die Beispiele und Abbildungen in diesem Thema basieren auf dieser Engine.

> **Note**
>
> Beim Serverstart prüft die Blusam-Laufzeit, ob einige obligatorische technische Tabellen vorhanden sind, und erstellt sie, falls sie nicht gefunden werden können. Folglich müssen der Rolle, die in der Konfiguration für den Zugriff auf die Blusam-Datenbank verwendet wurde, die Rechte zum Erstellen, Aktualisieren und Löschen der Datenbanktabellen (sowohl der Zeilen als auch der Tabellendefinitionen selbst) gewährt werden. Hinweise zur Deaktivierung von Blusam finden Sie unter. the section called “Blusam-Konfiguration”

Caching

Zusätzlich zum Speicher selbst arbeitet Blusam schneller, wenn es an eine Cache-Implementierung gekoppelt ist.

Derzeit werden zwei Cache-Engines EhCache und Redis unterstützt, von denen jede ihren eigenen Anwendungsfall hat:

- EhCache : Eigenständiger eingebetteter flüchtiger lokaler Cache
  - NICHT für die Bereitstellung einer verwalteten Umgebung mit AWS Mainframe-Modernisierung geeignet.
  - Wird in der Regel verwendet, wenn ein einzelner Knoten, z. B. ein einzelner Apache Tomcat-Server, für die Ausführung der modernisierten Anwendungen verwendet wird. Beispielsweise kann der Knoten für das Hosten von Batch-Job-Aufgaben vorgesehen sein.
  - Flüchtig: Die EhCache Cache-Instanz ist flüchtig; ihr Inhalt geht verloren, wenn der Server heruntergefahren wird.
  - Eingebettet: Der EhCache und der Server teilen sich denselben JVM-Speicherplatz (muss bei der Definition der Spezifikationen für den Hostcomputer berücksichtigt werden).
- Redis: Gemeinsamer persistenter Cache
  - Geeignet für die Bereitstellung einer verwalteten Umgebung im AWS Rahmen der Mainframe-Modernisierung.
  - Wird in der Regel in Situationen mit mehreren Knoten verwendet, insbesondere wenn sich mehrere Server hinter einem Load-Balancer befinden. Der Cache-Inhalt wird von allen Knoten gemeinsam genutzt.

- Redis ist persistent und nicht an die Lebenszyklen der Knoten gebunden. Es läuft auf einem eigenen Computer oder Dienst (z. B. Amazon ElastiCache). Der Cache ist von allen Knoten aus erreichbar.

#### Sperren

Um den gleichzeitigen Zugriff auf Datensätze und Aufzeichnungen zu bewältigen, setzt Blusam auf ein konfigurierbares Schließsystem. Das Sperren kann auf beide Ebenen angewendet werden: auf Datensätze und Datensätze:

- Das Sperren eines Datensatzes für Schreibzwecke verhindert, dass alle anderen Clients Schreiboperationen auf dieser Ebene (Datensatz oder Datensatz) ausführen.
- Das Sperren auf Datensatzebene für Schreibvorgänge verhindert, dass andere Clients Schreiboperationen nur für den angegebenen Datensatz ausführen.

Die Konfiguration des Blusam-Sperrsystems sollte entsprechend der Cache-Konfiguration erfolgen:

- Wenn diese EhCache Option als Cache-Implementierung ausgewählt wird, ist keine weitere Sperrkonfiguration erforderlich, da das standardmäßige In-Memory-Sperrsystem verwendet werden sollte.
- Wenn Redis als Cache-Implementierung ausgewählt wird, ist eine Redis-basierte Sperrkonfiguration erforderlich, um den gleichzeitigen Zugriff von mehreren Knoten aus zu ermöglichen. Der für Sperren verwendete Redis-Cache muss nicht derselbe sein wie der, der für Datensätze verwendet wird. Hinweise zur Konfiguration eines Redis-basierten Schließsystems finden Sie unter. the section called “Blusam-Konfiguration”

## Intrinsics von Blusam und Datenmigration aus älteren Versionen

#### Speichern von Datensätzen: Datensätze und Indizes

Jeder Legacy-Datensatz wird, wenn er in Blusam importiert wird, in einer speziellen Tabelle gespeichert. Jede Zeile der Tabelle steht für einen Datensatz mit zwei Spalten:

- Die numerische ID-Spalte vom Typ Big Integer, das ist der Primärschlüssel der Tabelle und wird verwendet, um die Relative Byte-Adresse (RBA) des Datensatzes zu speichern. Die RBA stellt den Offset in Byte vom Anfang des Datensatzes dar und beginnt bei 0.

- Die Byte-Array-Datensatzspalte, in der der Inhalt des Rohdatensatzes gespeichert wird.

Sehen Sie sich zum Beispiel den Inhalt eines KSDS-Datensatzes an, der in der CardDemo Anwendung verwendet wird:

```
1  SELECT * FROM public.aws_m2_carddemo_acctdata_vsam_ksds
2  ORDER BY id ASC
```

Data output | Messages | Notifications

| | id [PK] bigint | record bytea |
|---|---|---|
| 1 | 0 | [binary data] |
| 2 | 300 | [binary data] |
| 3 | 600 | [binary data] |
| 4 | 900 | [binary data] |
| 5 | 1200 | [binary data] |
| 6 | 1500 | [binary data] |
| 7 | 1800 | [binary data] |
| 8 | 2100 | [binary data] |
| 9 | 2400 | [binary data] |
| 10 | 2700 | [binary data] |
| 11 | 3000 | [binary data] |
| 12 | 3300 | [binary data] |
| 13 | 3600 | [binary data] |

- Dieser spezielle Datensatz hat Datensätze mit fester Länge, deren Länge 300 Byte beträgt (daher ist die Sammlung von IDs ein Vielfaches von 300).
- Standardmäßig zeigt das pgAdmin-Tool, das zur Abfrage von PostgreSQL-Datenbanken verwendet wird, keine Byte-Array-Spalteninhalte an, sondern gibt stattdessen ein [Binärdaten] -Label aus.
- Der Inhalt des Rohdatensatzes entspricht dem Rohdatensatz-Export aus der Legacy-Version, ohne dass eine Konvertierung erforderlich ist. Insbesondere findet keine Zeichensatzkonvertierung statt. Das bedeutet, dass alphanumerische Teile des Datensatzes von modernisierten Anwendungen unter Verwendung des alten Zeichensatzes, höchstwahrscheinlich einer EBCDIC-Variante, dekodiert werden müssen.

Was die Metadaten des Datensatzes und die Schlüsselindizes angeht: Jeder Datensatz ist mit zwei Zeilen in der genannten Tabelle verknüpft. `metadata` Dies ist die Standardbenennungskonvention. Informationen zur Anpassung finden Sie unterthe section called “Blusam-Konfiguration”.

| | name [PK] text | metadata oid |
|---|---|---|
| 1 | AWS_M2_CARDDEMO_ACCTDATA_VSAM_KSDS | 66320 |
| 2 | AWS_M2_CARDDEMO_ACCTDATA_VSAM_KSDS__internal | 66321 |

- Die erste Zeile hat den Datensatznamen als Wert der Namensspalte. Die Metadatenspalte ist eine binäre Spalte, die eine binäre Serialisierung der allgemeinen Metadaten des angegebenen Datensatzes enthält. Details hierzu finden Sie unter the section called “Allgemeine Metadatenattribute von Datensätzen”.
- Die zweite Zeile enthält den Datensatznamen mit dem Suffix `__internal'` als Wert der Namensspalte. Der binäre Inhalt der Metadatenspalte hängt vom „Gewicht“ des Datensatzes ab.
  - Bei kleinen/mittleren Datensätzen ist der Inhalt eine komprimierte Serialisierung von:
    - Definition der vom Datensatz verwendeten Schlüssel; die Primärschlüsseldefinition (für KSDS) und gegebenenfalls alternative Schlüsseldefinitionen (für KSDS/ESDS)
    - gegebenenfalls die Schlüsselindizes (KSDS/ESDS mit alternativen Schlüsseldefinitionen): werden für das indizierte Durchsuchen von Datensätzen verwendet; der Schlüsselindex ordnet der RBA eines Datensatzes einen Schlüsselwert zu;
    - Längenübersicht der Datensätze: wird für das sequentielle bzw. relative Durchsuchen von Datensätzen verwendet;
  - Bei großen/sehr großen Datensätzen handelt es sich bei dem Inhalt um eine komprimierte Serialisierung von:
    - Definition der vom Datensatz verwendeten Schlüssel; die Primärschlüsseldefinition (für KSDS) und gegebenenfalls alternative Schlüsseldefinitionen (für KSDS/ESDS)

Darüber hinaus werden Indizes für große/sehr große Datensätze (falls zutreffend) mithilfe eines Paginierungsmechanismus gespeichert; binäre Serialisierungen von Indexseiten werden als Zeilen einer eigenen Tabelle gespeichert (eine Tabelle pro Datensatzschlüssel). Jede Indexseite wird in einer Zeile mit den folgenden Spalten gespeichert:

- id: technische Kennung der Indexseite (numerischer Primärschlüssel);
- firstkey: Binärwert des ersten (niedrigsten) Schlüsselwerts, der auf der Indexseite gespeichert ist;
- lastkey: Binärwert des letzten (höchsten) Schlüsselwerts, der auf der Indexseite gespeichert ist;

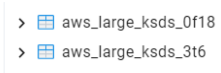
- Metadaten: binär komprimierte Serialisierung der Indexseite (Zuordnung von Schlüsselwerten zu Datensätzen). RBAs

| | id [PK] bigint | firstkey bytea | lastkey bytea | metadata oid |
|---|---|---|---|---|
| 1 | 1 | [binary data] | [binary da... | 6458928 |
| 2 | 2 | [binary data] | [binary da... | 6458929 |
| 3 | 3 | [binary data] | [binary da... | 6458930 |
| 4 | 4 | [binary data] | [binary da... | 6458931 |
| 5 | 5 | [binary data] | [binary da... | 6458932 |
| 6 | 6 | [binary data] | [binary da... | 6458933 |
| 7 | 7 | [binary data] | [binary da... | 6458934 |
| 8 | 8 | [binary data] | [binary da... | 6458935 |
| 9 | 9 | [binary data] | [binary da... | 6458936 |

Der Tabellenname ist eine Verkettung des Datensatznamens und des internen Schlüsselnamens. Er enthält Informationen über den Schlüssel, z. B. den Schlüsselversatz, ob der Schlüssel Duplikate akzeptiert (auf true gesetzt, um Duplikate zuzulassen) und die Schlüssellänge. Stellen Sie sich zum Beispiel einen Datensatz mit dem Namen "AWS_LARGE_KSDS“ vor, der die folgenden zwei definierten Schlüssel hat:

- Primärschlüssel [Offset: 0, Duplikate: falsch, Länge:18]
- Alternativschlüssel [Offset: 3, Duplikate: wahr, Länge: 6]

In diesem Fall speichern die folgenden Tabellen die Indizes, die sich auf die beiden Schlüssel beziehen.




Optimierung des I/O-Durchsatzes mithilfe des Write-Behind-Mechanismus

Um die Leistung von Einfüge-/Aktualisierungs- und Löschvorgängen zu optimieren, stützt sich die Blusam-Engine auf einen konfigurierbaren Write-Behind-Mechanismus. Der Mechanismus basiert auf einem Pool von dedizierten Threads, die Persistenzoperationen mithilfe von Abfragen für Massenaktualisierungen durchführen, um den I/O-Durchsatz für den Blusam-Speicher zu maximieren.

Die Blusam-Engine sammelt alle Aktualisierungsvorgänge, die von Anwendungen an Datensätzen ausgeführt werden, und erstellt Chargen von Datensätzen, die zur Bearbeitung an die dedizierten Threads weitergeleitet werden. Die Lose werden dann mithilfe von Massenaktualisierungsabfragen dauerhaft im Blusam-Speicher gespeichert, wodurch die Verwendung atomarer Persistenzoperationen vermieden wird und die bestmögliche Nutzung der Netzwerkbandbreite gewährleistet wird.

Der Mechanismus verwendet eine konfigurierbare Verzögerung (standardmäßig eine Sekunde) und eine konfigurierbare Losgröße (standardmäßig 10.000 Datensätze). Die Build-Persistenzabfragen werden ausgeführt, sobald die erste der beiden folgenden Bedingungen erfüllt ist:

- Die konfigurierte Verzögerung ist abgelaufen und das Los ist nicht leer
- Die Anzahl der Datensätze in dem zu behandelnden Los erreicht den konfigurierten Grenzwert

Informationen zur Konfiguration des Write-Behind-Mechanismus finden Sie unter. the section called “Optionale Eigenschaften”

Das richtige Speicherschema auswählen

Wie im vorherigen Abschnitt gezeigt, hängt die Art und Weise, wie Datensätze gespeichert werden, von ihrem „Gewicht“ ab. Aber was wird für einen Datensatz als klein, mittel oder groß angesehen? Wann sollte man sich für die paginierte Speicherstrategie entscheiden und nicht für die reguläre?

Die Antwort auf diese Frage hängt von den folgenden Faktoren ab.

- Die Menge des verfügbaren Speichers auf jedem der Server, auf denen die modernisierten Anwendungen gehostet werden, die diese Datensätze verwenden werden.
- Die Menge des verfügbaren Speichers in der Cache-Infrastruktur (falls vorhanden).

Bei Verwendung eines Speicherschemas für nicht paginierte Indizes werden beim Öffnen des Datensatzes die vollständigen Sammlungen mit Schlüsselindizes und Datensatzgrößen für jeden Datensatz in den Serverspeicher geladen. Wenn Caching erforderlich ist, können außerdem alle Datensatzdatensätze beim regulären Verfahren vorab in den Cache geladen werden, was zu einer Erschöpfung der Speicherressourcen auf der Seite der Cache-Infrastruktur führen kann.

Abhängig von der Anzahl der definierten Schlüssel, der Länge der Schlüsselwerte, der Anzahl der Datensätze und der Anzahl der gleichzeitig geöffneten Datensätze kann die Menge des verbrauchten Speichers für die gegebenen bekannten Anwendungsfälle grob berechnet werden.

Weitere Informationen hierzu finden Sie unter the section called “Schätzung des Speicherbedarfs für einen bestimmten Datensatz”.

Blusam-Migration

Sobald das richtige Speicherschema für einen bestimmten Datensatz ausgewählt wurde, muss der Blusam-Speicher durch die Migration älterer Datensätze aufgefüllt werden.

Um dies zu erreichen, muss man rohe Binärexporte der älteren Datensätze verwenden, ohne dass während des Exportvorgangs eine Zeichensatzkonvertierung verwendet wird. Achten Sie bei der Übertragung von Datensatz-Exporten aus dem Altsystem darauf, dass das Binärformat nicht beschädigt wird. Erzwingen Sie beispielsweise den Binärmodus, wenn Sie FTP verwenden.

Die rohen Binärexporte enthalten nur die Datensätze. Für den Importmechanismus ist es nicht erforderlich, keys/indexes exports as all keys/indexes dass sie im laufenden Betrieb durch den Importmechanismus neu berechnet werden.

Sobald ein binärer Export eines Datensatzes verfügbar ist, gibt es mehrere Optionen, ihn nach Blusam zu migrieren:

In der verwalteten AWS Umgebung von Mainframe Modernization:

- Importieren Sie Datensätze mithilfe der speziellen Funktion. Siehe the section called “Datensätze für Anwendungen importieren”.

or

- Verwenden Sie die Funktion zum Massenimport von Datensätzen. Siehe the section called “Referenz zur Definition von Datensätzen” und the section called “Beispiel für ein Datensatz-Anforderungsformat für VSAM”.

or

- Verwenden Sie ein Groovy-Skript, um Datensätze mithilfe spezieller Ladedienste zu importieren.

> **Note**
>
> Das Importieren von LargeKSDS und LargeESDS in verwalteten Umgebungen mit Mainframe Modernization ist derzeit nur mit Groovy-Skripten möglich.

Auf AWS Blu Age Runtime bei Amazon EC2:

- Importieren Sie den Datensatz mit demthe section called “Blusam Verwaltungskonsole”.

or

- Verwenden Sie ein Groovy-Skript, um Datensätze mithilfe spezieller Ladedienste zu importieren.

Importieren Sie Datensätze mithilfe von Groovy-Skripten

In diesem Abschnitt erfahren Sie, wie Sie Groovy-Skripte schreiben, um ältere Datensätze in Blusam zu importieren.

Es beginnt mit einigen obligatorischen Importen:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList; //used for alternate keys if any
```

Danach baut der Code für jeden zu importierenden Datensatz auf dem angegebenen Muster auf:

1. ein Kartenobjekt erstellen oder löschen
2. füllen Sie die Karte mit den erforderlichen Eigenschaften (dies variiert je nach Art des Datensatzes — Einzelheiten siehe unten)
3. ruft den richtigen Ladedienst ab, der für die Datensatzart in der Dienstregistrierung verwendet werden soll
4. führe den Dienst aus und verwende dabei die Map als Argument

Es gibt 5 Dienstimplementierungen, die mithilfe der folgenden Bezeichner aus der Dienstregistrierung abgerufen werden können:

- `"BluesamKSDSFileLoader"`: für kleine/mittelgroße KSDS
- `"BluesamESDSFileLoader"`für kleine/mittelgroße ESDS
- `"BluesamRRDSFileLoader"`: für RRDS
- `"BluesamLargeKSDSFileLoader"`: für große KSDS

- `"BluesamLargeESDSFileLoader"`: für große ESDS

Ob Sie die reguläre oder die große Version des Dienstes für KSDS/ESDS wählen, hängt von der Größe der Datensätze und der Speicherstrategie ab, die Sie dafür anwenden möchten. Informationen zur Auswahl der richtigen Speicherstrategie finden Sie unter. the section called “Das richtige Speicherschema auswählen”

Um den Datensatz erfolgreich in Blusam importieren zu können, müssen dem Ladedienst die richtigen Eigenschaften zur Verfügung gestellt werden.

Allgemeine Eigenschaften:

- Obligatorisch (für alle Arten von Datensätzen)
  - „BlueSamManager“: Der erwartete Wert ist `applicationContext.getBean(BluesamManager.class)`
  - „DatasetName“: Name des Datensatzes als Zeichenfolge
  - "inFilePath": Pfad zum Export des Legacy-Datensatzes als Zeichenfolge
  - „RecordLength“: Die feste Datensatzlänge oder 0 für einen Datensatz mit variabler Datensatzlänge als Ganzzahl
- Optional
  - Wird für große Datensätze nicht unterstützt:
    - „isAppend“: ein boolesches Flag, das angibt, dass der Import im Anfügemodus erfolgt (Datensätze werden an einen vorhandenen Blusam-Datensatz angehängt).
    - „useCompression“: ein boolesches Flag, das angibt, dass die Komprimierung zum Speichern von Metadaten verwendet wird.
  - Nur für große Datensätze:
    - "indexingPageSizeInMb": die Größe jeder Indexseite in Megabyte für jeden der Schlüssel des Datensatzes als strikt positive Ganzzahl

Eigenschaften, die von der Art des Datensatzes abhängen:

- KSDS/großes KSDS:
  - mandatory
    - „PrimaryKey“: Die Primärschlüsseldefinition, die einen Konstruktoraufruf verwendet. `com.netfective.bluage.gapwalk.bluesam.metadata.Key`

  - fakultativ:
    - „alternateKeys“: eine Liste (`java.util.List`) von alternativen Schlüsseldefinitionen, die mithilfe von `com.netfective.bluage.gapwalk.bluesam.metadata.Key` Konstruktoraufrufen erstellt wurde.
- ESDS/Große ESDS:
  - fakultativ:
    - „alternateKeys“: eine Liste (`java.util.List`) von alternativen Schlüsseldefinitionen, die mithilfe von `com.netfective.bluage.gapwalk.bluesam.metadata.Key` Konstruktoraufrufen erstellt wurde.
- ROTE ZAHLEN:
  - keiner.

Wichtige Konstruktoraufrufe:

- `new Key(int offset, int length)`: erstellt ein Key-Objekt mit bestimmten Schlüsselattributen (Offset und Länge) und erlaubt keine Duplikate. Diese Variante sollte verwendet werden, um einen Primärschlüssel zu definieren.
- `new Key(boolean allowDuplicates, int offset, int length)`: erstellt ein Key-Objekt mit bestimmten Schlüsselattributen (Offset und Länge) und Duplikaten, die ein Flag zulassen.

Die folgenden Groovy-Beispiele veranschaulichen verschiedene Ladeszenarien.

Es wird ein großes KSDS mit zwei alternativen Schlüsseln geladen:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;

// Loading a large KSDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "largeKsdsSample");
map.put("inFilePath", "/work/samples/largeKsdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
ArrayList altKeys = [new Key(true, 10, 8), new Key(false, 0, 9)]
```

```
map.put("alternateKeys", altKeys);
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamLargeKSDSFileLoader");
service.runService(map);
```

Laden eines ESDS mit variabler Datensatzlänge ohne alternative Schlüssel:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry

// Loading an ESDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "esdsSample");
map.put("inFilePath", "/work/samples/esdsSampleExport");
map.put("recordLength", 0);
def service = ServiceRegistry.getService("BluesamESDSFileLoader");
service.runService(map);
```

Exporte von Datensätzen mit variabler Datensatzlänge enthalten die obligatorischen RDW-Informationen (Record Decriptor Word), damit Datensätze beim Lesen aufgeteilt werden können.

Laden eines RRDS mit fester Datensatzlänge:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry

// Loading a RRDS into Blusam
def map = [:]
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", "rrdsSample");
map.put("inFilePath", "/work/samples/rrdsSampleExport");
map.put("recordLength", 180);
def service = ServiceRegistry.getService("BluesamRRDSFileLoader");
service.runService(map);
```

Laden von Datensätzen im Multischema-Modus:

Multischemamodus: In einigen älteren Systemen sind VSAM-Dateien in Dateisätzen organisiert, sodass Programme auf Daten innerhalb bestimmter Partitionen zugreifen und diese ändern können.

Moderne Systeme behandeln jeden Dateisatz als Schema und ermöglichen so eine ähnliche Datenpartitionierung und Zugriffskontrolle.

Informationen zum Aktivieren des Multischemamodus in der `application-main.yml` Datei finden Sie unter. the section called “Blusam-Konfiguration” In diesem Modus können Datensätze mithilfe eines Shared Contexts, einer speicherinternen Registrierung für Laufzeitinformationen, in ein bestimmtes Schema geladen werden. Um einen Datensatz in ein bestimmtes Schema zu laden, stellen Sie dem Datensatznamen den entsprechenden Schemanamen voran.

Laden einer KSDS-Datei in ein bestimmtes Schema für den Multischemamodus:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;
import com.netfective.bluage.gapwalk.rt.shared.SharedContext;

// Loading a KSDS into Blusam
def map = [:]
String schema = "schema1";
String datasetName = schema+"|"+"ksdsSample";
SharedContext.get().setCurrentBlusamSchema(schema);
schema = SharedContext.get().getCurrentBlusamSchema();
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", datasetName);
map.put("inFilePath", "/work/samples/ksdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamKSDSFileLoader");
service.runService(map);
```

Laden einer großen KSDS-Datei in ein bestimmtes Schema für den Multischemamodus:

```
import com.netfective.bluage.gapwalk.bluesam.BluesamManager
import com.netfective.bluage.gapwalk.bluesam.metadata.Key;
import com.netfective.bluage.gapwalk.rt.provider.ServiceRegistry
import java.util.ArrayList;
import com.netfective.bluage.gapwalk.rt.shared.SharedContext;

// Loading a Large KSDS into Blusam
def map = [:]
```

```
String schema = "schema1";
String datasetName = schema+"|"+"largeKsdsSample";
SharedContext.get().setCurrentBlusamSchema(schema);
schema = SharedContext.get().getCurrentBlusamSchema();
map.put("bluesamManager", applicationContext.getBean(BluesamManager.class));
map.put("datasetName", datasetName);
map.put("inFilePath", "/work/samples/LargeKsdsSampleExport");
map.put("recordLength", 49);
map.put("primaryKey", new Key(0, 18));
map.put("indexingPageSizeInMb", 25);
def service = ServiceRegistry.getService("BluesamLargeKSDSFileLoader");
service.runService(map);
```

Darüber hinaus kann ein Konfigurationseintrag (der in der `application-main.yml` Konfigurationsdatei festgelegt werden muss) zur Feinabstimmung des Importvorgangs verwendet werden:

- `bluesam.fileLoading.commitInterval`: eine strikt positive Ganzzahl, die das Commit-Intervall für den regulären ESDS/KSDS/RRDS Importmechanismus definiert. Gilt nicht für Importe großer Datensätze. Der Standardwert ist 100000.

## Blusam-Konfiguration

Die Konfiguration von Blusam erfolgt in der `application-main.yml` Konfigurationsdatei (oder in der `application-bac.yml` Konfigurationsdatei für die eigenständige Bereitstellung der Blusam Administration Console — BAC — Anwendung).

Blusam muss in zweierlei Hinsicht konfiguriert werden:

- Konfiguration des Blusam-Speichers und des Zugriffs auf Caches
- Konfiguration der Blusam-Engine

### Konfiguration des Blusam-Speichers und des Cache-Zugriffs

Informationen zur Konfiguration des Zugriffs auf Blusam-Speicher und -Caches mithilfe von Secrets-Managern oder Datenquellen finden Sie unter. the section called “AWS Blu Age Runtime-Konfiguration”

> **Note**
>
> In Bezug auf den Zugriff auf den Blusam-Speicher verweisen die verwendeten Anmeldeinformationen auf eine Verbindungsrolle mit den entsprechenden Rechten. Damit die Blusam-Engine wie erwartet funktionieren kann, muss die Verbindungsrolle über die folgenden Rechte verfügen:

- stellen Sie eine Verbindung zur Datenbank her
- Tabellen und Ansichten erstellen/löschen/ändern/kürzen
- Zeilen in Tabellen und Ansichten auswählen/einfügen/löschen/aktualisieren
- Funktionen oder Prozeduren ausführen

Konfiguration der Blusam-Engine

Blusam-Unterstützung wird deaktiviert

Lassen Sie uns zunächst erwähnen, dass es möglich ist, die Blusam-Unterstützung vollständig zu deaktivieren, indem Sie die Eigenschaft auf setzen. `bluesam.disabled true` Beim Start der Anwendung wird in den Serverprotokollen eine Informationsmeldung angezeigt, die an die Deaktivierung von Blusam erinnert:

```
BLUESAM is disabled. No operations allowed.
```

In diesem Fall ist keine weitere Konfiguration von Blusam erforderlich, und jeder Versuch, Blusam-bezogene Funktionen zu verwenden (entweder programmgesteuert oder über REST-Aufrufe), führt bei der Ausführung des Java-Codes zu einer Meldung mit einer entsprechenden `UnsupportedOperationException` Erklärung, dass Blusam deaktiviert wurde.

Eigenschaften der Blusam-Engine

Die Konfigurationseigenschaften der Blusam-Engine sind unter dem Bluesam-Schlüsselpräfix neu gruppiert:

Obligatorische Eigenschaften

- `cache`: muss mit der ausgewählten Cache-Implementierung bewertet werden. Gültige Werte für sind:
  - `ehcache`: Für die lokale Verwendung von eingebettetem Ehcache. Die entsprechenden Einschränkungen für Anwendungsfälle finden Sie weiter oben.
  - `redis`: Für die gemeinsame Remote-Nutzung des Redis-Caches. Dies ist die bevorzugte Option für den verwalteten AWS Anwendungsfall Mainframe-Modernisierung.
  - `none`: Um das Speicher-Caching zu deaktivieren
- `persistence`: mit pgsql zu bewerten (PostgreSQL-Engine: Minimalversion 10.0 — empfohlene Version >=14.0
- Datenquellenreferenz: verweist auf `<persistence engine>.dataSource` die DataSource-Definition für die Verbindung zum Blusam-Speicher, die an anderer Stelle in der Konfigurationsdatei definiert ist. Üblicherweise wird es benannt. `bluesamDs`

> **Note**
>
> Immer wenn Redis als Cache-Mechanismus verwendet wird, entweder für Daten oder für Sperren (siehe unten), muss der Zugriff auf die Redis-Instanzen konfiguriert werden. Details hierzu finden Sie unter the section called “Verfügbare Redis-Cache-Eigenschaften”.

Optionale Eigenschaften

Blusam Locks: Den Eigenschaften wird das Präfix vorangestellt `locks`

- `cache`: Der einzig verwendbare Wert ist die Angabe`redis`, dass der Redis-basierte Sperrmechanismus verwendet wird (der auch verwendet werden sollte, wenn der Blusam-Storage-Cache auf Redis basiert). Wenn die Eigenschaft fehlt oder nicht auf gesetzt ist`redis`, wird stattdessen der standardmäßige In-Memory-Sperrmechanismus verwendet.
- `lockTimeOut`: ein positiver Wert vom Typ Long Integer, der die Zeitüberschreitung in Millisekunden angibt, bevor ein Versuch, ein bereits gesperrtes Element zu sperren, als fehlgeschlagen markiert wird. Die Standardeinstellung ist. `500`
- `locksDeadTime`: Ein positiver langer Integer-Wert, der die maximale Zeit, ausgedrückt in Millisekunden, darstellt, für die eine Anwendung gesperrt werden kann. Sperren werden automatisch als abgelaufen markiert und nach Ablauf dieser Zeit wieder freigegeben. Die Standardeinstellung ist; `1000`

- `locksCheck`: eine Zeichenfolge, die verwendet wird, um die vom aktuellen Blusam Lock Manager verwendete Strategie zur Sperrüberprüfung zu definieren, die sich mit dem Entfernen abgelaufener Sperren befasst. Soll aus den folgenden Werten ausgewählt werden:
  - `off`: Es werden keine Prüfungen durchgeführt. Wir raten davon ab, da tote Sperren auftreten können.
  - `reboot`: Die Prüfungen werden beim Neustart oder beim Start der Anwendung durchgeführt. Alle abgelaufenen Sperren werden zu diesem Zeitpunkt freigegeben. Dies ist die Standardeinstellung.
  - `timeout`: Prüfungen werden beim Neustart oder beim Start der Anwendung durchgeführt oder wenn beim Versuch, einen Datensatz zu sperren, ein Timeout abläuft. Abgelaufene Sperren werden sofort aufgehoben.

Write-Behind-Mechanismus: Den Eigenschaften wird ein Schlüssel vorangestellt: `write-behind`

- `enabled`: `true` (Standard- und empfohlener Wert) oder`false`, um den Write-Behind-Mechanismus zu aktivieren oder zu deaktivieren. Die Deaktivierung des Mechanismus beeinträchtigt die Schreibleistung erheblich und es wird davon abgeraten.
- `maxDelay`: eine maximale Dauer, für die die Threads ausgelöst werden sollen. Der Standardwert ist `"1s"` (eine Sekunde). Es ist generell eine gute Idee, den Standardwert beizubehalten, es sei denn, dieser Wert muss unter bestimmten Bedingungen angepasst werden. In jedem Fall sollte der Wert niedrig gehalten werden (unter 3 Sekunden). Das Format für die Verzögerungszeichenfolge `<time unit>` ist: `<integer value><optional whitespace><time unit>` wobei aus den folgenden Werten ausgewählt werden soll:
  - `"ns"`: Nanosekunden
  - `"µs"`: Mikrosekunden
  - `"ms"`: Millisekunden
  - `"s"`: Sekunden
- `threads`: die Anzahl der dedizierten Write-Behind-Threads. Die Standardeinstellung ist. 5 Sie müssen diesen Wert an die Rechenleistung des Hosts anpassen, auf dem die Blusam-Engine ausgeführt wird. Es ist nicht relevant, einen viel höheren Wert zu verwenden, in der Hoffnung auf eine Leistungssteigerung, da der limitierende Faktor die Fähigkeit des Speicher-RDBMS sein wird, zahlreiche gleichzeitige Batch-Abfragen zu verarbeiten. Die empfohlenen Werte liegen normalerweise im Bereich von 4 bis 8.

- `batchSize`: eine positive Ganzzahl, die die maximale Anzahl von Datensätzen in einem Los angibt, die zur Sammelverarbeitung an einen Thread versandt werden. Ihr Wert muss zwischen 1 und 32767 liegen. Die Standardeinstellung ist. `10000` Die Verwendung `1` als Wert macht den Zweck des Mechanismus zunichte, der darin besteht, die Verwendung atomarer Aktualisierungsabfragen zu vermeiden. Der geeignete Mindestwert, den man verwenden kann, ist ungefähr. `1000`

Integrierte EhCache Feinabstimmung: Den Eigenschaften wird ein Schlüssel vorangestellt: `ehcache`

- `resource-pool`:
  - `size`: Größe des zugewiesenen Speichers für den eingebetteten Cache, ausgedrückt als Zeichenfolge. Der Standardwert ist `"1024MB"` (1 Gigabyte). Muss im Hinblick auf den verfügbaren Speicher des Rechners, der die Blusam-Engine hostet, und die Größe der von der Anwendung verwendeten Datensätze angepasst werden. Das Format der Größenzeichenfolge ist: `<integer value><optional whitespace><memory unit>` wobei `<memory-unit>` aus den folgenden Werten ausgewählt werden soll:
    - `B`:Byte
    - `KB`: Kilobyte
    - `MB`: Megabyte
    - `GB`: Gigabyte
    - `TB`: Terabyte
  - `heap`: `true` oder`false`, um anzugeben, ob der Cache JVM-Heap-Speicher verbraucht oder nicht. Der Standardwert ist `true` (schnellste Option für die Cache-Leistung, aber der Cache-Speicher verbraucht Speicher aus dem JVM-On-Heap-RAM-Speicher). Wenn Sie diese Eigenschaft auf setzen, `false` wird zum Off-Heap-Speicher gewechselt, der aufgrund des erforderlichen Austauschs mit dem JVM-Heap langsamer ist.
- `timeToLiveMillis`: Die Dauer (in Millisekunden), für die ein Cache-Eintrag im Cache verbleibt, bevor er als abgelaufen betrachtet und entfernt wird. Wenn diese Eigenschaft nicht angegeben ist, laufen Cache-Einträge standardmäßig nicht automatisch ab.

Eigenschaften der Konfiguration mehrerer Schemas

- `multiSchema`: false (Standardwert) oder true, um den Multischemamodus für Blusam zu deaktivieren oder zu aktivieren — verfügbar ab Version 4.4.0.
- `pgsql`:

- `schemas`: Eine Liste von Schemanamen, die die Anwendung im Multischema-Modus für Blusam verwenden wird.
- `fallbackSchema`: Der Name des Fallback-Schemas für die Verwendung im Multischemamodus. Wenn ein Datensatz im aktuellen Schemakontext nicht gefunden wird, wird dieses Schema für BLUSAM-bezogene Operationen an diesem Datensatz verwendet.

Beispiel für einen Konfigurationsausschnitt:

```
dataSource:
  bluesamDs:
    driver-class-name: org.postgresql.Driver
    ...
    ...
bluesam:
  locks:
    lockTimeOut: 700
  cache: ehcache
  persistence: pgsql
  ehcache:
    resource-pool:
      size: 8GB
  write-behind:
    enabled: true
    threads: 8
    batchsize: 5000
  pgsql:
    dataSource : bluesamDs
```

Beispiel für einen Konfigurationsausschnitt (mit aktiviertem Multi-Schema-Modus für Blusam):

```
dataSource:
  bluesamDs:
    driver-class-name: org.postgresql.Driver
    ...
    ...
bluesam:
  locks:
    lockTimeOut: 700
  cache: ehcache
  persistence: pgsql
  ehcache:
```

```
    resource-pool:
      size: 8GB
  write-behind:
    enabled: true
    threads: 8
    batchsize: 5000
  multiSchema: true
  pgsql:
    dataSource : bluesamDs
    schemas:
      - "schema1"
      - "schema2"
      - "schema3"
    fallbackSchema: schema3
```

> **Note**
>
> Blusam-Metadatenschemas, einschließlich der in der `application-main.yml` Datei für den Multischemamodus aufgelisteten Schemas, werden in der Blusam-Datenbank erstellt, wenn sie nicht existieren und der Benutzer über ausreichende Rechte verfügt.

## Blusam Administrationskonsole

Die Blusam Administration Console (BAC) ist eine Webanwendung, die zur Verwaltung des Blusam-Speichers verwendet wird. Informationen zum BAC finden Sie unter. the section called “Blusam Verwaltungskonsole”

## Anhang

### Allgemeine Metadatenattribute von Datensätzen

Liste der allgemeinen Attribute für die Serialisierung von Datensatz-Metadaten:

- Name (des Datensatzes)
- Typ (KSDS, LargeKSDS, ESDS, LargeESDS oder RRDS)
- Cache-Aufwärmkennzeichen (ob der Datensatz beim Serverstart vorab in den Cache geladen werden soll oder nicht)
- Kennzeichen für die Verwendung der Komprimierung (ob Datensätze in einem komprimierten oder unformatierten Format gespeichert werden sollen)

- Erstellungsdatum
- Datum der letzten Änderung
- Datensatzkennzeichnung mit fester Länge (ob die Datensätze alle dieselbe Länge haben oder nicht)
- Datensatzlänge — nur sinnvoll für eine feste Datensatzlänge
- Seitengröße (wird verwendet, um die paginierten SQL-Abfragen anzupassen, die bei Bedarf zum Vorladen des Caches verwendet werden)
- Größe (Größe des Datensatzes — kumulierte Länge der Datensätze)
- letzter Offset (Offset, d. h. RBA des zuletzt zum Datensatz hinzugefügten Datensatzes)
- nächster Offset (nächster verfügbarer Offset zum Hinzufügen eines neuen Datensatzes zum Datensatz)
- falls sinnvoll, Definition der vom Datensatz verwendeten Schlüssel; jeder Schlüssel wird durch seine Art (primär oder Teil der alternativen Schlüsselsammlung) und durch drei Attribute definiert:
  - Offset: Position des Startbytes des Schlüsselwerts im Datensatz;
  - Länge: Länge des Schlüsselwerts in Byte. Somit ist der Schlüsselwert das Byte-Array, das die Teilmenge des Datensatzes darstellt, der an der Position beginnt `key offset` und an dieser `key offset + length - 1` endet;
  - Markierung für erlaubte Duplikate: ob der Schlüssel Duplikate akzeptiert oder nicht (auf true gesetzt, um Duplikate zuzulassen).

Schätzung des Speicherbedarfs für einen bestimmten Datensatz

Bei kleinen bis mittelgroßen Datensätzen werden die Metadaten (Größen und Indizes für verschiedene Schlüssel) vollständig in den Speicher geladen. Um die richtigen Ressourcen für die Maschine zuzuweisen, auf der der Server gehostet wird, auf dem modernisierte Anwendungen ausgeführt werden, muss der Speicherverbrauch ermittelt werden, der durch die Blusam-Datensätze verursacht wird, insbesondere in Bezug auf Metadaten. In diesem Abschnitt werden den betroffenen Betreibern praktische Antworten gegeben.

Die angegebenen Formeln gelten nur für kleine bis mittlere Blusam-Datensätze, nicht für die Speicherstrategie „Large“.

Metadaten von Blusam-Datensätzen

Bei einem Blusam-Datensatz werden die Metadaten in zwei Teile aufgeteilt:

- Kernmetadaten: enthalten globale Informationen über den Datensatz. Der damit verbundene Speicherbedarf kann im Vergleich zu den internen Metadaten als vernachlässigbar angesehen werden.
- interne Metadaten: enthalten Informationen über die Größe der Datensätze und die wichtigsten Indizes. Wenn ein Datensatz nicht leer ist, verbraucht dieser Speicherplatz, wenn er auf den Anwendungsserver geladen wird, der modernisierte Anwendungen hostet. In den folgenden Abschnitten wird detailliert beschrieben, wie der verbrauchte Speicher mit der Anzahl der Datensätze zunimmt.

Berechnung des internen Metadaten-Footprints

Karte mit den Größen der Datensätze

Zunächst speichern die internen Metadaten eine Map, die die Größe jedes Datensatzes (als Ganzzahl) anhand seiner RBA (relative Byte-Adresse — gespeichert als lange Zahl) enthält.

Der Speicherbedarf dieser Datenstruktur beträgt in Byte: `80 * number of records`

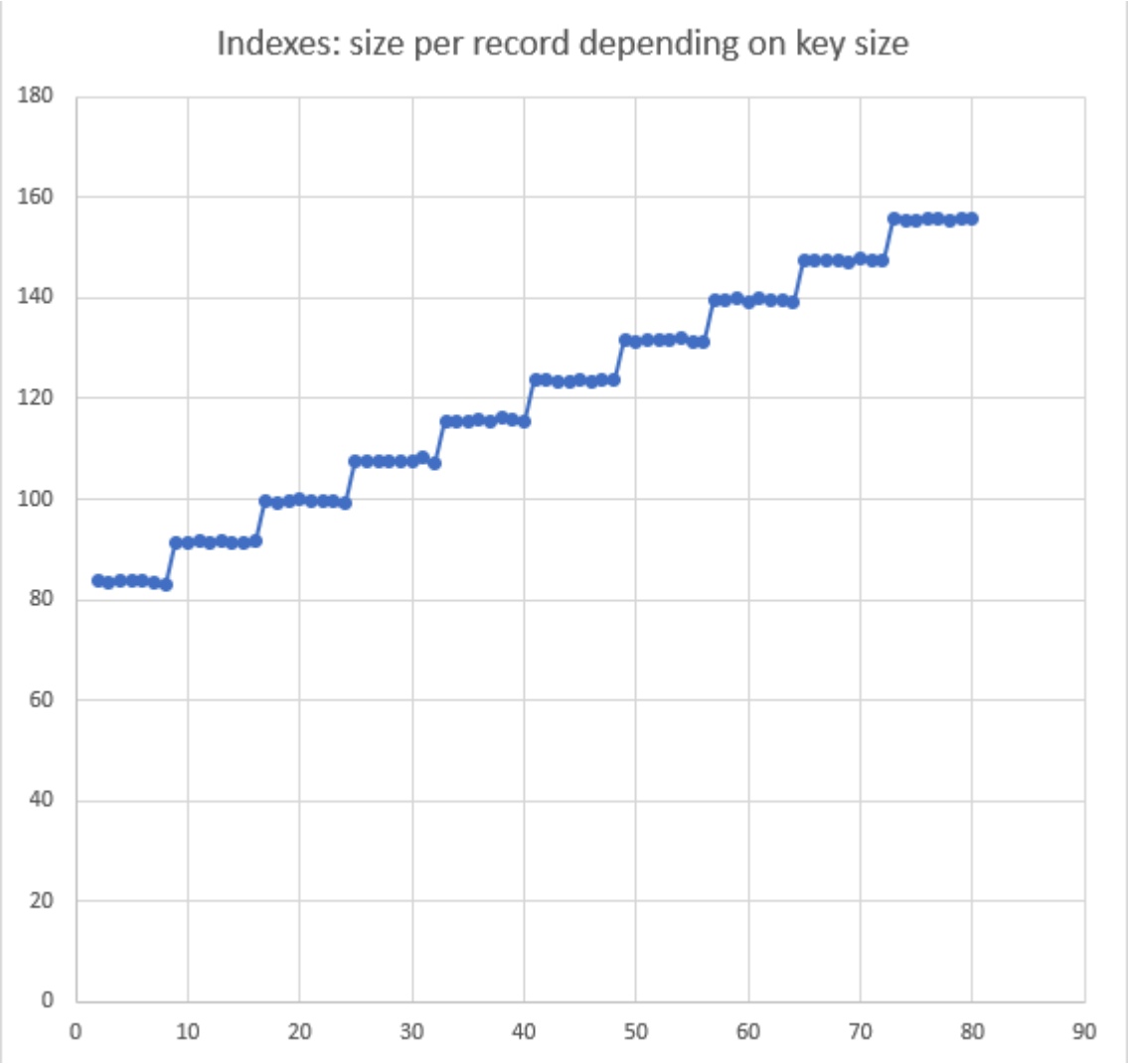
Dies gilt für alle Arten von Datensätzen.

Indizes

Was die Indizes für den Primärschlüssel von KSDS oder die alternativen Schlüssel sowohl für ESDS als auch für KSDS angeht, hängt die Berechnung des Footprints von zwei Faktoren ab:

- die Anzahl der Datensätze im Datensatz;
- die Größe des Schlüssels in Byte.

Die folgende Grafik zeigt die Größe des Schlüsselindexes pro Datensatz (Y-Achse) basierend auf der Größe des Schlüssels (X-Achse).




Die entsprechende Formel zur Berechnung des Footprints für einen bestimmten Schlüsselindex eines Datensatzes lautet:

```
index footprint = number of records * ( 83 + 8 (key length / 8))
```

wobei '/' für die Ganzzahldivision steht.

Beispiele:

- Datensatz 1:
  - Anzahl der Datensätze = 459 996
  - Schlüssellänge = 15 also (Schlüssellänge/8) = 1
  - Index-Footprint = 459 996 * (83 + (8*1)) = 41 859 636 Byte (= ca. 39 MB)

- Datensatz 2:
  - Anzahl der Datensätze = 13 095 783
  - Schlüssellänge = 18 also (Schlüssellänge/8) = 2
  - Index-Footprint = 13 095 783 * (83 + (8*2)) = 1 296 482 517 Byte (= ca. 1,2 GB)

Der gesamte Footprint für einen bestimmten Datensatz ist die Summe aller Footprints für alle Schlüsselindizes und des Footprints für die Größentabelle der Datensätze.

Nehmen wir zum Beispiel den Beispieldatensatz 2, der nur einen einzigen Schlüssel enthält, so lautet der globale Footprint:

- Karte mit den Größen der Datensätze: 13 095 783 * 80 = 1 047 662 640 Byte
- Wichtige Indizes: 1 296 482 517 Byte (siehe oben)
- Gesamtspeicherplatz = 2 344 145 157 Byte (= ca. 2,18 GB)

## Verfügbare Programme in der Utility-Webanwendung

Die Utility-Webanwendung unterstützt verschiedene ältere Plattform-Hilfsprogramme wie IDCAMS, INFUTILB, SORT usw. Informationen zum Konfigurieren des Zugriffs auf die Anwendung finden Sie unter. Zugriff auf Dienstprogramme für verwaltete Anwendungen konfigurieren

Liste der Programme

- JCLBCICS-Hilfsprogramm- Wird von Batch verwendet, um den Status des Bluesam-Datensatzes auf zu setzen. open/enabled or closed/disabled

### JCLBCICS-Hilfsprogramm

JCLBCICS ist ein JCL-Hilfsprogramm, das entwickelt wurde, um den Bluesam-Datensatz auf den open/enabled or closed/disabled. An open/enabled status will block access to the dataset from batch programs while a closed/disabled Status zu setzen, dass der Datensatz für den Zugriff auf JICS-Onlinedienste nicht verfügbar ist.

Usage

- JCLBCICS ändert die STATUS-Spalte in der Jics-Tabelle FILE_TABLE und die OPEN_STATUS-Spalte in der Bluesam BLUESAM_STATUS-Tabelle basierend auf der Groovy-Konfiguration für den DD-Namen.

```
.open(ddName) -> ENABLED in Jics FILE_TABLE table, OPEN in Bluesam BLUESAM_STATUS
 table
.close(ddName) -> DISABLED in Jics FILE_TABLE table, CLOSED in Bluesam BLUESAM_STATUS
 table
```

- Die Größe des `application-utility-pgm.yml` DD-Namens ist global in der Konfigurationsdatei konfigurierbar.

```
jclbcics.ddname.size: 7
```

- Die globale DD-Namensgröße kann in einem einzelnen Schritt überschrieben werden, indem die überschriebene Größe mit den folgenden Zeilen in Groovy angegeben wird und dann StepParams als Parameter für diesen Schritt verwendet werden.

```
TreeMap stepMapTransfo = [:]
Map stepParams = ["MapTransfo":stepMapTransfo]
stepParams["MapTransfo"]["JCLBCICS_OVERRIDDEN_SIZE"] = '7'
...
.withParameters(stepParams)
.runProgram("JCLBCICS")
```

- Bei der Einstellung der DD-Namensgröße beträgt die maximale effektive DD-Namensgröße 8.
- Wenn die DDName-Länge größer als die angegebene DD-Namensgröße ist, wird sie bis zum Ende gekürzt, sodass sie der Größe des DD-Namens entspricht.
- Platzhalter werden in DDName unterstützt. Wenn ein Sternchen (*) an das Ende von DDName angehängt wird oder die Länge des DDNamens weniger als 8 beträgt.

```
.open("DTSNAME*")
```

Beispiel-Code

```
// DD name with overridden size of 7 bytes
 def stepSTEP007(Object shell, Map params, Map programResults) {
   shell.with {
     if (checkValidProgramResults(programResults)) {
```

```
        TreeMap stepMapTransfo = [:]
        Map stepParams = ["MapTransfo":stepMapTransfo]
        stepParams["MapTransfo"]["JCLBCICS_OVERRIDDEN_SIZE"] = '7'
        return execStep("STEP007", "JCLBCICS", programResults, {
          mpr
            .withDatasetsConfiguration(new DatasetsConfiguration()
            .close("DTSNAME"))
            .withParameters(stepParams)
            .runProgram("JCLBCICS")
         })
      }
    }
}
```

# AWS Blu Age Blusam Verwaltungskonsole

Die Blusam Administration Console (BAC) ist eine sichere Webanwendung für den Umgang mit Blusam-Datensätzen. Dieses Handbuch behandelt die BAC-Benutzeroberfläche. Informationen zur Fernverwaltung über REST-Endpunkte finden Sie unter. the section called “REST-Endpunkte der Blusam-Anwendungskonsole ”

Themen


- Bereitstellen des BAC
- Verwenden des BAC
- LISTCAT-JSON-Format


## Bereitstellen des BAC

Der BAC ist als sichere einzelne Webanwendung verfügbar, die das Webarchivformat (.war) verwendet. Es soll zusammen mit der BluAge Gapwalk-Anwendung auf einem Apache Tomcat-Anwendungsserver bereitgestellt werden, kann aber auch als eigenständige Anwendung bereitgestellt werden. Der BAC erbt den Zugriff auf den Blusam-Speicher von der Gapwalk-Application-Konfiguration, falls vorhanden.

Das BAC hat eine eigene Konfigurationsdatei mit dem Namen. `application-bac.yml` Einzelheiten zur Konfiguration finden Sie unterthe section called “Dedizierte BAC-Konfigurationsdatei”.

Das BAC ist gesichert. Einzelheiten zur Sicherheitskonfiguration finden Sie unterthe section called “Konfiguration der Sicherheit für den BAC”.

Dedizierte BAC-Konfigurationsdatei

Eigenständige Bereitstellung: Wenn der BAC allein mit der Gapwalk-Anwendung bereitgestellt wird, muss die Verbindung zum Blusam-Speicher in der Konfigurationsdatei application-bac.yml konfiguriert werden.

Standardwerte für die Datensatzkonfiguration, die zum Durchsuchen von Datensätzen verwendet wird, müssen in der Konfigurationsdatei festgelegt werden. Siehe the section called “Datensätze aus einem Datensatz durchsuchen”. Auf der Seite zum Durchsuchen von Datensätzen kann ein optionaler Maskenmechanismus verwendet werden, der es ermöglicht, eine strukturierte Ansicht des Inhalts eines Datensatzes anzuzeigen. Einige Eigenschaften wirken sich auf die Ansicht von Datensätzen aus, wenn Masken verwendet werden.

Die folgenden konfigurierbaren Eigenschaften müssen in der Konfigurationsdatei festgelegt werden. Die BAC-Anwendung geht von keinem Standardwert für diese Eigenschaften aus.

| Schlüssel | Typ | Beschreibung |
|---|---|---|
| `bac.crud.limit` | Ganzzahl | Ein positiver Ganzzahlwert, der die maximale Anzahl von Datensätzen angibt, die beim Durchsuchen von Datensätzen zurückgegeben werden. `0`Benutzen bedeutet unbegrenzt. Empfohlener Wert: `10` (passen Sie dann den Wert von Datensatz zu Datensatz auf der Browserse ite an, um ihn an Ihre Bedürfnisse anzupassen). |
| `bac.crud.encoding` | Zeichenfolge | Der Standardname des Zeichensatzes, der zum Dekodieren von Datensatz bytes als alphanumerischen Inhalt verwendet wird. Der |

| Schlüssel | Typ | Beschreibung |
| --- | --- | --- |
| | | angegebene Zeichensatzname muss Java-kompatibel sein (die unterstützten Zeichensätze finden Sie in der Java-Dokumentation ). Empfohlener Wert: der Legacy-Zeichensatz, der auf der Legacy-Plattform verwendet wird, aus der die Datensätze stammen. In den meisten Fällen wird es sich dabei um eine EBCDIC-Variante handeln. |
| `bac.crud.initCharacter` | Zeichenfolge | Das Standardzeichen (Byte), das zur Initialisierung von Datenelementen verwendet wird. Zwei Sonderwerte können verwendet werden: `"LOW-VALUE"` das 0x00-Byte (empfohlener Wert) und `"HI-VALUE"` das 0xFF-Byte. Wird verwendet, wenn Masken angewendet werden. |
| `bac.crud.defaultCharacter` | Zeichenfolge | Das Standardzeichen (Byte) als Zeichenfolge aus einem Zeichen, das zum Auffüllen von Datensätzen verwendet wird (rechts). Empfohlener Wert: `" "` (Leerzeichen). Wird verwendet, wenn Masken angewendet werden. |

| Schlüssel | Typ | Beschreibung |
| --- | --- | --- |
| `bac.crud.blankCharacter` | Zeichenfolge | Das Standardzeichen (Byte) als Zeichenfolge aus einem Zeichen, das zur Darstellung von Leerzeichen in Datensätzen verwendet wird. Empfohlener Wert: " " (Leerzeichen). Wird verwendet, wenn Masken angewendet werden. |
| `bac.crud.strictZoned` | boolesch | Eine Markierung, die angibt, welcher Zonenmodus für den Datensatz verwendet wird. Wenn`true`, wird der Modus Strict Zone verwendet; wenn`false`, wird der Modus Modified Zoned verwendet. Empfohlener Wert:`true`. Wird verwendet, wenn Masken aufgetragen werden. |
| `bac.crud.decimalSeparator` | Zeichenfolge | Das Zeichen, das als Dezimaltrennzeichen in numerisch bearbeiteten Feldern verwendet wird (wird verwendet, wenn Masken angewendet werden). |

| Schlüssel | Typ | Beschreibung |
| --- | --- | --- |
| `bac.crud.currencySign` | Zeichenfolge | Das Standardzeichen als einstellige Zeichenfolge, das zur Darstellung von Währungen in numerisch bearbeiteten Feldern verwendet wird, wenn Formatierung angewendet wird (wird verwendet, wenn Masken angewendet werden). |
| `bac.crud.pictureCurrencySign` | Zeichenfolge | Das Standardzeichen in Form einer aus einem Zeichen bestehenden Zeichenfolge zur Darstellung von Währungen in Bildern mit numerisch bearbeiteten Feldern (wird verwendet, wenn Masken angewendet werden). |

Das folgende Beispiel ist ein Ausschnitt aus einer Konfigurationsdatei.

```
bac.crud.limit: 10
bac.crud.encoding: ascii
bac.crud.initCharacter: "LOW-VALUE"
bac.crud.defaultCharacter: " "
bac.crud.blankCharacter: " "
bac.crud.strictZoned: true
bac.crud.decimalSeparator: "."
bac.crud.currencySign: "$"
bac.crud.pictureCurrencySign: "$"
```

Konfiguration der Sicherheit für den BAC

Die Konfiguration der Sicherheit für den BAC basiert auf den Mechanismen, die auf dieser Dokumentationsseite beschrieben werden. Das Authentifizierungsschema ist OAuth2, und die Konfigurationsdetails für Amazon Cognito oder Keycloak werden bereitgestellt.

Das allgemeine Setup kann zwar angewendet werden, einige Einzelheiten zum BAC müssen hier jedoch detailliert beschrieben werden. Der Zugriff auf die BAC-Funktionen ist durch eine rollenbasierte Richtlinie geschützt und basiert auf den folgenden Rollen.

- ROLE_USER:
  - Grundlegende Benutzerrolle
  - Import, Export, Erstellung oder Löschung von Datensätzen sind nicht erlaubt
  - Keine Kontrolle über Caching-Richtlinien
  - Keine Verwaltungsfunktionen erlaubt
- ROLE_ADMIN:
  - Erbt die ROLE_USER-Berechtigungen
  - Alle Datensatzoperationen sind zulässig
  - Verwaltung von Caching-Richtlinien zulässig

Installation der Masken

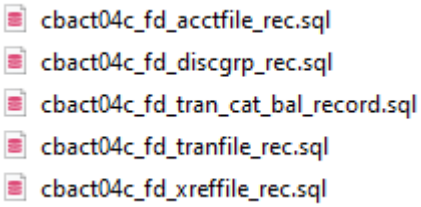
Im Blusam-Speicher werden Datensätze aus Gründen der Vielseitigkeit und Leistung in einer Byte-Array-Spalte in der Datenbank gespeichert. Eine praktische Funktion des BAC ist der Zugriff auf eine strukturierte Ansicht der Geschäftsdatensätze anhand von Feldern, die aus Sicht der Anwendung erfolgt. Dies stützt sich auf die SQL-Masken, die während des intensiven BluAge Modernisierungsprozesses erstellt wurden.

Damit die SQL-Masken generiert werden können, stellen Sie bitte sicher, dass Sie die entsprechende Option (`export.SQL.masks`) in der Konfiguration des BluInsights Transformation Centers auf true setzen:




Die Masken sind Teil der Modernisierungsartefakte, die BluInsights für ein bestimmtes Projekt heruntergeladen werden können. Es handelt sich um SQL-Skripte, die in modernisierten Programmen organisiert sind und die anwendungsorientierte Sicht auf Datensätze vermitteln.

Wenn Sie beispielsweise die CardDemo AWS-Beispielanwendung verwenden, finden Sie in den heruntergeladenen Artefakten aus dem Modernisierungsergebnis dieser Anwendung die folgenden SQL-Masken für das Programm cbact04C.cbl:




Jeder SQL-Maskenname ist die Verkettung des Programmnamens und des Datensatzstrukturnamens für einen bestimmten Datensatz innerhalb des Programms.

Schauen wir uns zum Beispiel das Programm [Cbact04c.cbl] an, der angegebene Dateisteuerungseintrag:

```
    FILE-CONTROL.
        SELECT TCATBAL-FILE ASSIGN TO TCATBALF
               ORGANIZATION IS INDEXED
               ACCESS MODE  IS SEQUENTIAL
               RECORD KEY   IS FD-TRAN-CAT-KEY
               FILE STATUS  IS TCATBALF-STATUS.
```

ist der angegebenen FD-Datensatzdefinition zugeordnet

```
       FILE SECTION.
       FD  TCATBAL-FILE.
       01  FD-TRAN-CAT-BAL-RECORD.
           05 FD-TRAN-CAT-KEY.
              10 FD-TRANCAT-ACCT-ID             PIC 9(11).
              10 FD-TRANCAT-TYPE-CD             PIC X(02).
              10 FD-TRANCAT-CD                  PIC 9(04).
           05 FD-FD-TRAN-CAT-DATA               PIC X(33).
```

Die angegebene passende SQL-Maske `cbact04c_fd_tran_cat_bal_record.SQL` ist die Maske, die den Standpunkt des Programms Cbact04c.CBL auf den genannten FD-Datensatz wiedergibt. `FD-TRAN-CAT-BAL-RECORD`

Ihr Inhalt ist:

```
-- Generated by Blu Age Velocity
-- Mask : cbact04c_fd_tran_cat_bal_record

INSERT INTO mask (name, length) VALUES ('cbact04c_fd_tran_cat_bal_record', 50);
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_trancat_acct_id', 1, 11, false, 'zoned', 'integerSize=11!fractionalSize=0!
signed=false', (SELECT MAX(id) FROM mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_trancat_type_cd', 12, 2, false, 'alphanumeric', 'length=2', (SELECT MAX(id) FROM
 mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk)
 VALUES ('fd_trancat_cd', 14, 4, false, 'zoned', 'integerSize=4!fractionalSize=0!
signed=false', (SELECT MAX(id) FROM mask));
  INSERT INTO mask_item (name, c_offset, length, skip, type, options, mask_fk) VALUES
 ('fd_fd_tran_cat_data', 18, 33, false, 'alphanumeric', 'length=33', (SELECT MAX(id)
 FROM mask));
```

Masken werden im Blusam-Speicher mithilfe von zwei Tabellen gespeichert:

- Maske: wird zur Identifizierung von Masken verwendet. Die Spalten der Masstabelle sind:
  - Name: wird zum Speichern der Maskenidentifikation verwendet (wird als Primärschlüssel verwendet, muss also eindeutig sein)
  - Länge: Größe der Datensatzmaske in Byte

- mask_item: Wird zum Speichern von Maskendetails verwendet. Jedes elementare Feld aus einer FD-Datensatzdefinition erzeugt eine Zeile in der Tabelle mask_item mit Details zur Interpretation des angegebenen Datensatzteils. Die Spalten der Tabelle mask_item sind:
  - Name: Name des Datensatzfeldes, basierend auf dem Elementarnamen, wobei Kleinbuchstaben verwendet und der Bindestrich durch einen Unterstrich ersetzt wird
  - c_offset: Auf 1 basierender Offset des Datensatzunterteils, der für den Feldinhalt verwendet wird
  - Länge: Länge des Datensatzunterteils, der für den Feldinhalt verwendet wird, in Byte
  - skip: Markierung, die angibt, ob der angegebene Datensatzteil in der View-Präsentation übersprungen werden soll oder nicht
  - Typ: die Feldart (basierend auf der alten Picture-Klausel)
  - Optionen: zusätzliche Typoptionen — typabhängig
  - mask_fk: Verweis auf die Masken-ID, an die dieses Element angehängt werden soll

Beachten Sie Folgendes:

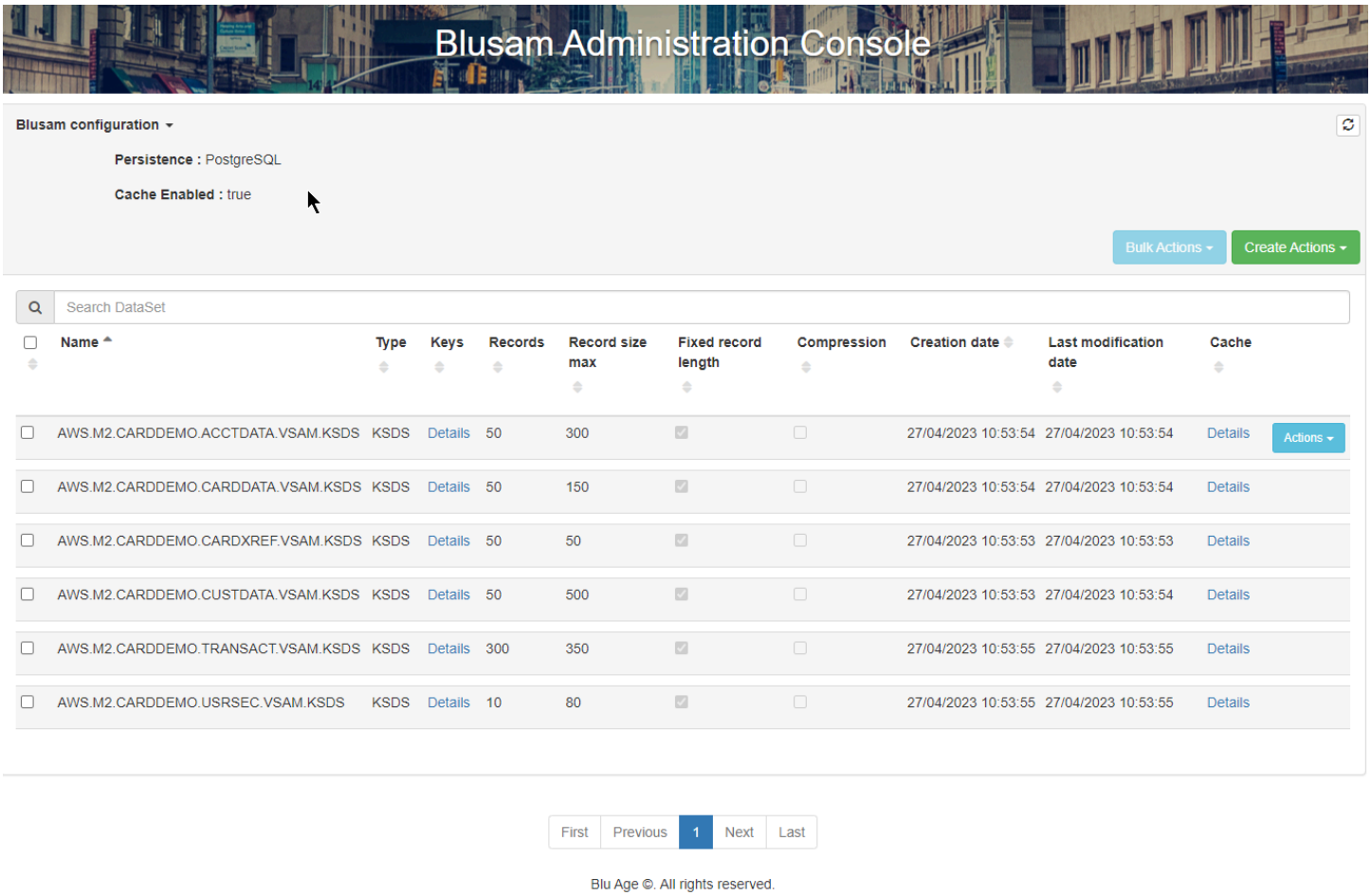
- SQL-Masken stellen die Sichtweise eines Programms auf Datensätze aus einem Datensatz dar: Verschiedene Programme haben möglicherweise unterschiedliche Ansichten zu einem bestimmten Datensatz. Installieren Sie nur die Masken, die Sie für Ihren Zweck als relevant erachten.
- Eine SQL-Maske kann auch die Sicht eines Programms repräsentieren, das auf einer 01-Datenstruktur aus dem Abschnitt WORKING STORAGE basiert, nicht nur aus einem FD-Datensatz. Die SQL-Masken sind entsprechend ihrer Art in Unterordner unterteilt:
  - Masken, die auf FD-Datensätzen basieren, befinden sich im Unterordner mit dem Namen `file`
  - 01 Masken, die auf Datenstrukturen basieren, befinden sich im Unterordner mit dem Namen `working`

  Die Definitionen von FD-Datensätzen stimmen zwar immer mit dem Inhalt eines Datensatzes überein, aber 01-Datenstrukturen sind möglicherweise nicht aufeinander abgestimmt oder stellen nur eine Teilmenge aus einem Datensatzdatensatz dar. Bevor Sie sie verwenden, überprüfen Sie den Code und machen Sie sich mit den möglichen Unzulänglichkeiten vertraut.

## Verwenden des BAC

Da der BAC gesichert ist und je nach Benutzerrolle Berechtigungen zur Nutzung von Funktionen erteilt, müssen Sie sich zunächst authentifizieren, um auf die Anwendung zuzugreifen. Nach dem

Authentifizierungsschritt werden Sie zur Startseite weitergeleitet. Auf der Startseite finden Sie die paginierte Liste der Datensätze, die sich im Blusam-Speicher befinden:




Um zur Startseite mit der Liste der Datensätze zurückzukehren, wählen Sie das Blu Age-Logo in der oberen linken Ecke einer beliebigen Seite der Anwendung. Das folgende Bild zeigt das Logo.




Der faltbare Header mit der Bezeichnung "BluSam Konfiguration“ enthält Informationen über die verwendete BluSam Speicherkonfiguration:

- `Persistence`: die persistente Speicher-Engine (PostgreSQL)
- `Cache Enabled`: ob der Speichercache aktiviert ist

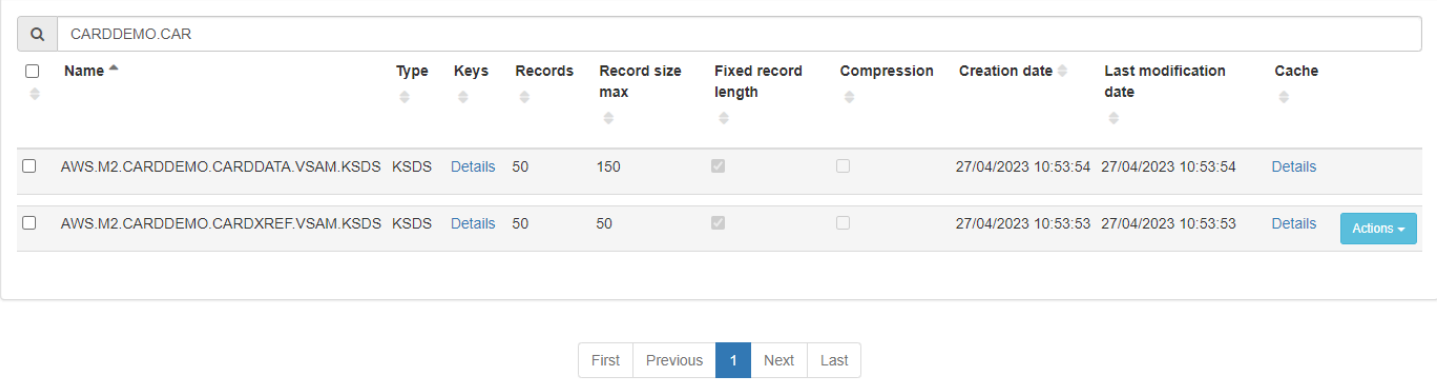
Auf der rechten Seite der Kopfzeile befinden sich zwei Dropdownlisten, in denen jeweils Operationen im Zusammenhang mit Datensätzen aufgeführt sind:

- Massen-Aktionen
- Aktionen erstellen

Weitere Informationen zum detaillierten Inhalt dieser Listen finden Sie unterthe section called “Bestehende Datensatzoperationen”.

Die Schaltfläche „Massenaktionen“ ist deaktiviert, wenn kein Datensatz ausgewählt wurde.

Sie können das Suchfeld verwenden, um die Liste anhand der Namen der Datensätze zu filtern:




Die folgende paginierte Liste zeigt einen Datensatz pro Tabellenzeile mit den folgenden Spalten:

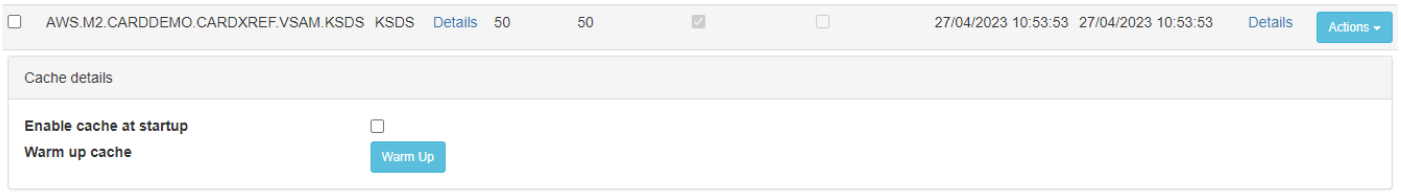
- Auswahl-Checkbox: Eine Checkbox zur Auswahl des aktuellen Datensatzes.
- Name: Der Name des Datensatzes.
- Typ: Der Typ des Datensatzes, einer der folgenden:
  - KSDS
  - ESDS
  - ROTE
- Schlüssel: Ein Link zum Ein- oder Ausblenden von Details zu den Schlüsseln (falls vorhanden). Zum Beispiel hat das angegebene KSDS den obligatorischen Primärschlüssel und einen alternativen Schlüssel.



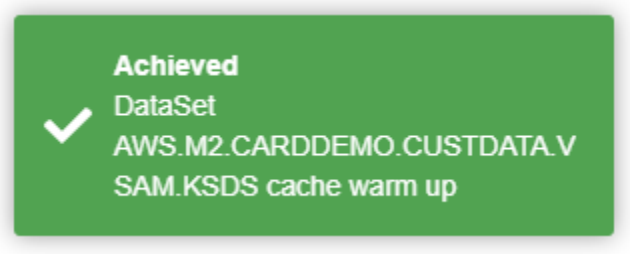
Pro Schlüssel gibt es eine Zeile mit den folgenden Spalten. Keines der Felder kann bearbeitet werden.

- Art des Schlüssels: entweder ein Primärschlüssel oder ein alternativer Schlüssel
- Name: der Name des Schlüssels
- Eindeutig: ob der Schlüssel doppelte Einträge akzeptiert
- Offset: Offset des Schlüsselbeginns innerhalb des Datensatzes
- Länge: Länge des Schlüsselteils im Datensatz in Byte

- Datensätze: Die Gesamtzahl der Datensätze im Datensatz.
- Max. Datensatzgröße: Die maximale Größe für Datensätze, ausgedrückt in Byte.
- Feste Datensatzlänge: Ein Kontrollkästchen, das angibt, ob die Datensätze eine feste Länge (ausgewählt) oder eine variable Länge (nicht ausgewählt) haben.
- Komprimierung: Ein Kontrollkästchen, das angibt, ob die Komprimierung auf gespeicherte Indizes angewendet (ausgewählt) oder nicht (nicht ausgewählt) wird.
- Erstellungsdatum: Das Datum, an dem der Datensatz im Blusam-Speicher erstellt wurde.
- Datum der letzten Änderung: Das Datum, an dem der Datensatz zuletzt im Blusam-Speicher aktualisiert wurde.
- Cache: Ein Link zum Ein- oder Ausblenden von Details zur Caching-Strategie, die auf diesen Datensatz angewendet wurde.




  - Cache beim Start aktivieren: Ein Kontrollkästchen, um die Start-Caching-Strategie für diesen Datensatz anzugeben. Wenn diese Option ausgewählt ist, wird der Datensatz beim Start in den Cache geladen.
  - Cache aufwärmen: Eine Schaltfläche, um den angegebenen Datensatz sofort in den Cache zu laden (das Hydratisieren des Caches dauert jedoch einige Zeit, abhängig von der Größe des Datensatzes und der Anzahl der Schlüssel). Nachdem der Datensatz in den Cache geladen wurde, erscheint eine Benachrichtigung wie die folgende.

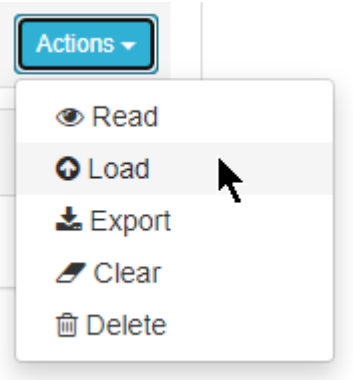



- Aktionen: Eine Dropdownliste mit möglichen Datensatzoperationen. Details hierzu finden Sie unter the section called “Bestehende Datensatzoperationen”.

Am Ende der Seite befindet sich ein reguläres Navigations-Widget mit Seiten, mit dem Sie die Seiten der Datensatzliste durchsuchen können.

Bestehende Datensatzoperationen

Für jeden Datensatz in der paginierten Liste gibt es eine Dropdownliste mit Aktionen mit folgendem Inhalt:



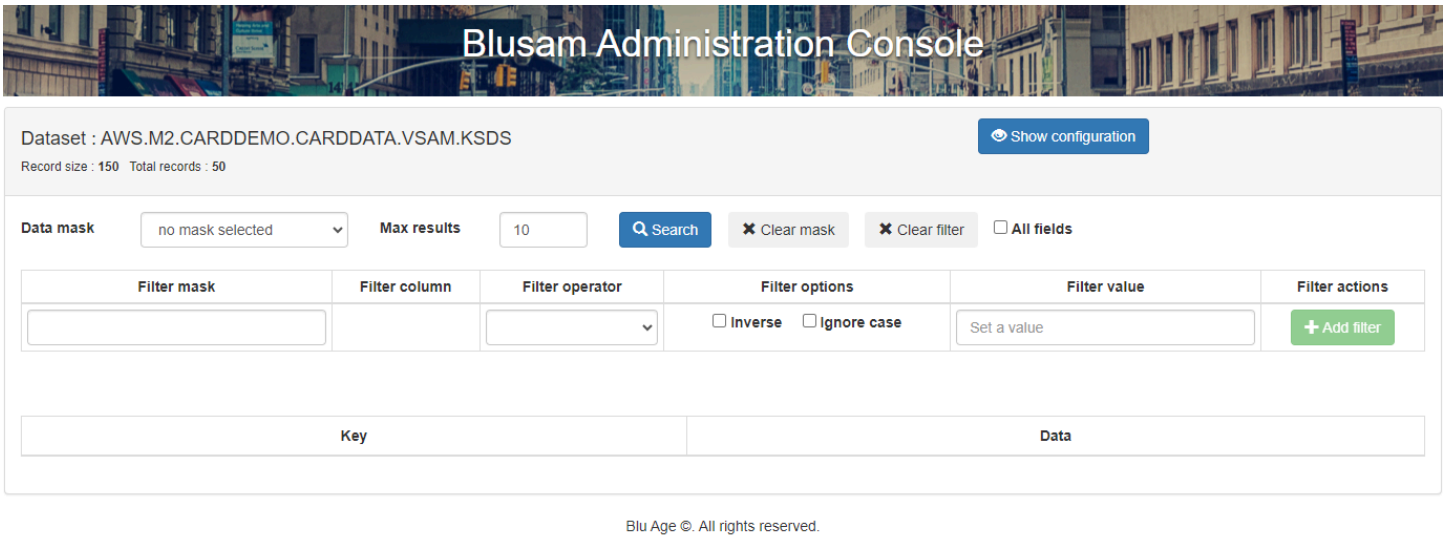

Jedes Element in der Liste ist ein aktiver Link, der es ermöglicht, die angegebene Aktion für den Datensatz auszuführen:

- Lesen: Datensätze aus den Datensätzen durchsuchen
- Laden: Importiert Datensätze aus einer älteren Datensatzdatei
- Export: Datensätze in eine Flat-Datei exportieren (kompatibel mit älteren Systemen)
- Löschen: entfernt alle Datensätze aus dem Datensatz
- Löschen: Löscht den Datensatz aus dem Speicher

Einzelheiten zu den einzelnen Aktionen finden Sie in den folgenden Abschnitten.

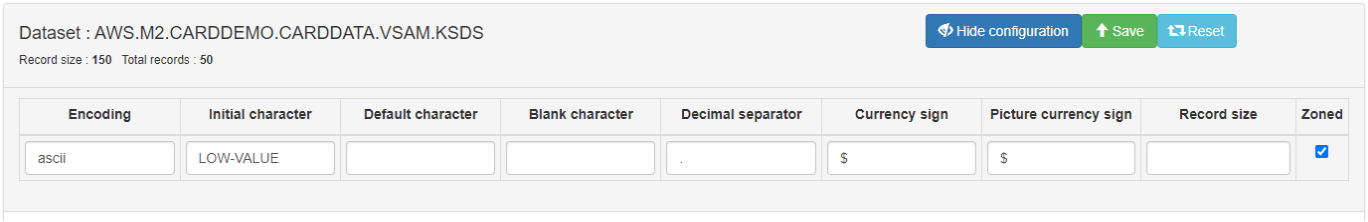
## Datensätze aus einem Datensatz durchsuchen

Wenn Sie die Aktion Lesen für einen bestimmten Datensatz wählen, wird die folgende Seite angezeigt.




Die Seite besteht aus:

- eine Kopfzeile mit:
  - Datensatz: der Name des Datensatzes
  - Datensatzgröße: Die feste Datensatzlänge, ausgedrückt in Byte
  - Datensätze insgesamt: Die Gesamtzahl der für diesen Datensatz gespeicherten Datensätze
  - Konfigurationsschaltfläche anzeigen (auf der rechten Seite): Eine Umschaltfläche zum Ein-/Ausblenden der Datensatzkonfiguration. Die Konfiguration ist zunächst ausgeblendet. Wenn Sie die Schaltfläche verwenden, sehen Sie die Konfiguration, wie in der folgenden Abbildung gezeigt.

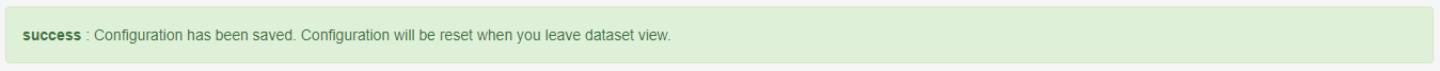



    Wenn die Konfiguration angezeigt wird, gibt es zwei neue Schaltflächen: Speichern und Zurücksetzen, die jeweils für Folgendes verwendet werden:

    - speichert die Konfiguration für diesen Datensatz und die aktuelle Arbeitssitzung
    - setzt die Konfiguration für alle Felder auf die Standardwerte zurück.

- Eine Liste konfigurierbarer Eigenschaften, um das Surferlebnis auf den jeweiligen Datensatz zuzuschneiden.

Die konfigurierbaren Eigenschaften entsprechen den unter beschriebenen Konfigurationseigenschaftenthe section called “Dedizierte BAC-Konfigurationsdatei”. In diesem Abschnitt finden Sie Informationen zur Bedeutung der einzelnen Spalten und zu den jeweiligen Werten. Jeder Wert kann hier für den Datensatz neu definiert und für die Arbeitssitzung gespeichert werden (mit der Schaltfläche Speichern). Nachdem Sie die Konfiguration gespeichert haben, erscheint ein Banner, das dem in der folgenden Abbildung ähnelt.




Das Banner gibt an, dass die Arbeitssitzung endet, wenn Sie die aktuelle Seite verlassen.

Es gibt eine zusätzliche konfigurierbare Eigenschaft, die im Abschnitt Konfiguration nicht dokumentiert ist: Datensatzgröße. Dies wird verwendet, um eine bestimmte Datensatzgröße, ausgedrückt in Byte, anzugeben, wodurch die für diesen Datensatz geltenden Masken gefiltert werden: Nur Masken, deren Gesamtlänge der angegebenen Datensatzgröße entspricht, werden in der Dropdownliste Datenmaske aufgeführt.

Das Abrufen von Datensätzen aus dem Datensatz wird durch die Schaltfläche Suchen ausgelöst, wobei alle Optionen und Filter in der Nähe verwendet werden.

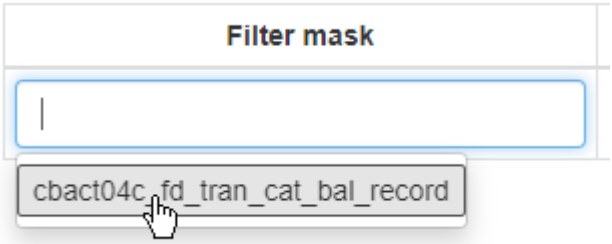
Erste Zeile mit Optionen:

- In der Dropdownliste Datenmaske werden die zutreffenden Masken angezeigt (unter Berücksichtigung der Datensatzgröße). Bitte beachten Sie, dass die Anpassung an die Datensatzgröße nicht ausreicht, um eine wirksame Maske zu sein. Die Maskendefinition muss auch mit dem Inhalt des Datensatzes kompatibel sein. Die hier ausgewählte Datenmaske hat
- Maximale Anzahl von Ergebnissen: Beschränkt die Anzahl der bei der Suche abgerufenen Datensätze. Auf 0 setzen, um unbegrenzt viele (paginierte) Ergebnisse aus dem Datensatz zu erhalten.
- Suchschaltfläche: Startet den Abruf von Datensätzen mithilfe von Filtern und Optionen
- Schaltfläche „Maske löschen“: Löscht die verwendete Maske, falls vorhanden, und wechselt auf der Ergebnisseite zurück zu einer Rohschlüssel-/Datenpräsentation.
- Schaltfläche „Filter löschen“: Löscht die verwendeten Filter, falls vorhanden, und aktualisiert die Ergebnisseite entsprechend.

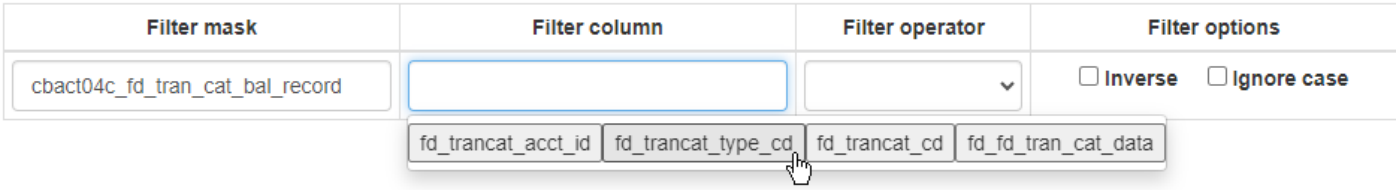
- Alle Felder umschalten: Wenn diese Option ausgewählt ist, werden Maskenelemente, die mit definiert `skip = true` sind, trotzdem angezeigt, andernfalls `skip = true` werden Maskenelemente mit ausgeblendet.

Nächste Filterreihen: Es ist möglich, eine Liste von Filtern zu definieren, die auf der Verwendung von Filterbedingungen basiert, die auf Felder (Spalten) einer bestimmten Maske angewendet werden, wie in der folgenden Abbildung gezeigt.

- Filtermaske: Der Name der Maske, aus der die Filterspalte ausgewählt werden soll. Wenn Sie das Feld auswählen, wird die Liste der zutreffenden Masken angezeigt. Sie können die gewünschte Maske aus dieser Liste auswählen.



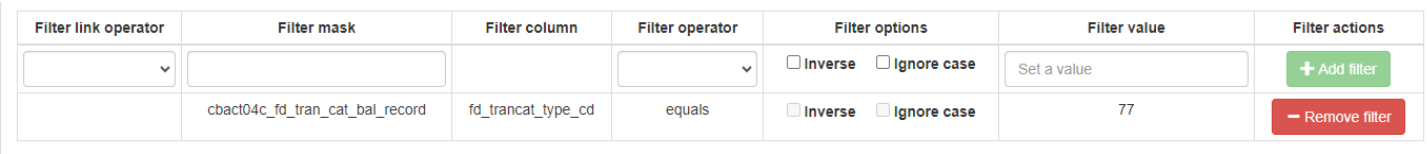

- Filterspalte: Der Name des Felds (der Spalte) aus der Maske, das zum Filtern von Datensätzen verwendet wird. Wenn Sie das Feld auswählen, wird die Liste der Maskenspalten angezeigt. Um das Feld Filterspalte auszufüllen, wählen Sie die gewünschte Zelle aus.




- Filteroperator: Ein Operator, der auf die ausgewählte Spalte angewendet werden soll. Die folgenden Operatoren sind verfügbar.
  - entspricht: Der Spaltenwert für den Datensatz muss dem Filterwert entsprechen
  - beginnt mit: Der Spaltenwert für den Datensatz muss mit dem Filterwert beginnen
  - endet mit: Der Spaltenwert für den Datensatz muss mit dem Filterwert enden
  - enthält: Der Spaltenwert für den Datensatz muss den Filterwert enthalten
- Filteroptionen:
  - Umgekehrt: wendet die umgekehrte Bedingung für den Filteroperator an. Beispielsweise wird „ist gleich“ durch „ist nicht gleich“ ersetzt;

  - Groß- und Kleinschreibung ignorieren: Bei alphanumerischen Vergleichen für den Filteroperator wird die Groß- und Kleinschreibung ignoriert
- Filterwert: Der Wert, der vom Filteroperator für den Vergleich mit der Filterspalte verwendet wird.

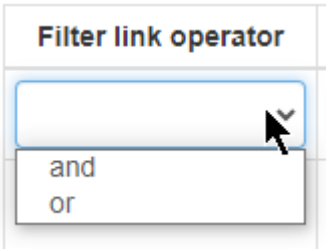
Sobald die Mindestanzahl von Filterelementen festgelegt ist (mindestens: Filtermaske, Filterspalte, Filteroperator und Filterwert müssen festgelegt sein), wird die Schaltfläche Filter hinzufügen aktiviert, und wenn Sie darauf klicken, wird eine neue Filterbedingung für die abgerufenen Datensätze erstellt. Eine weitere leere Zeile mit Filterbedingungen wird oben hinzugefügt, und die hinzugefügte Filterbedingung verfügt über die Schaltfläche Filter entfernen, mit der die angegebene Filterbedingung unterdrückt werden kann:




Wenn Sie die Suche starten, werden die gefilterten Ergebnisse in einer paginierten Tabelle angezeigt.

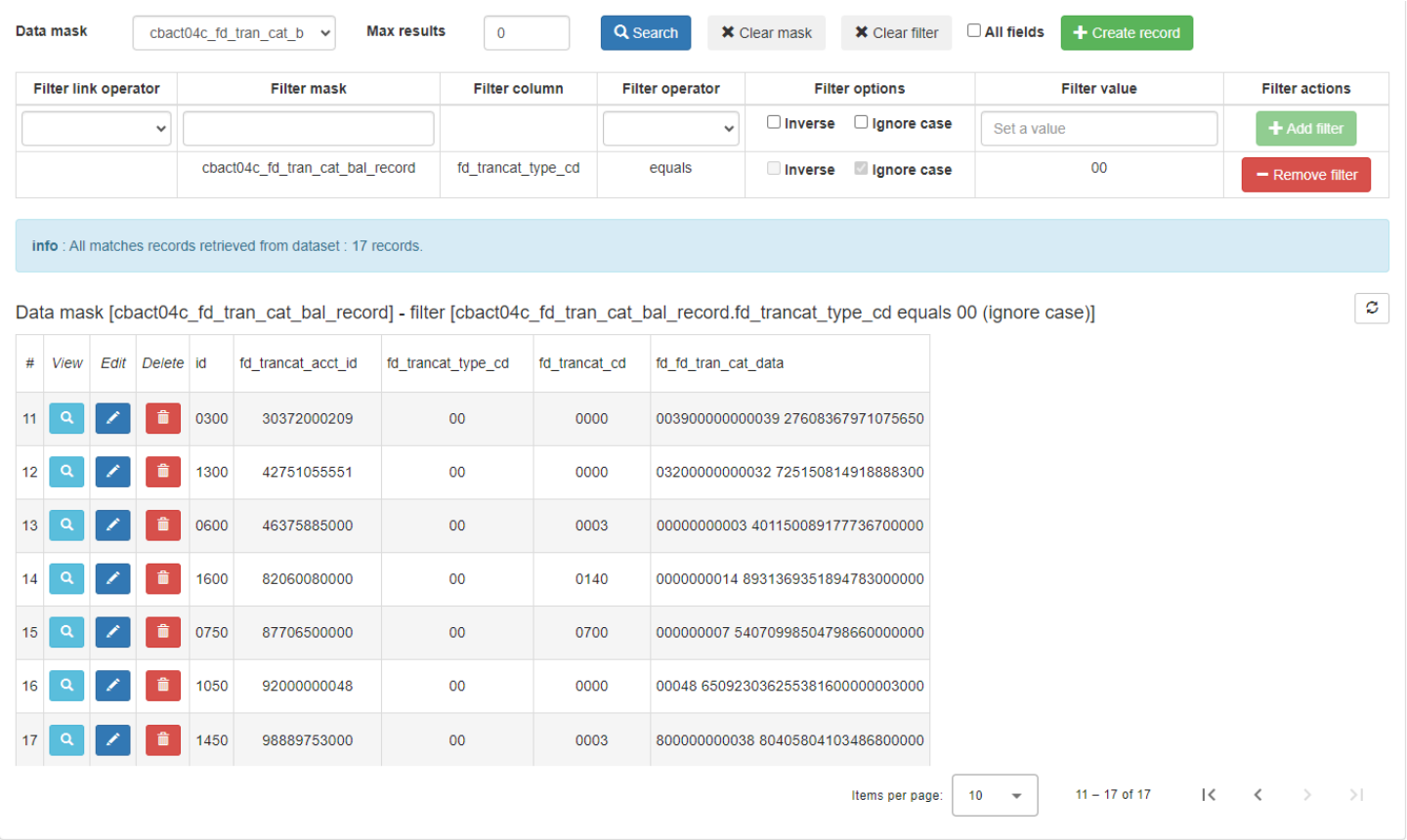
Hinweis

- Aufeinanderfolgende Filter sind durch ein Und oder ein Oder verknüpft. Jede neue Filterdefinition beginnt mit der Einstellung des Link-Operators, wie in der folgenden Abbildung dargestellt.




- Möglicherweise gibt es keine Datensätze, die den angegebenen Filterbedingungen entsprechen.

Andernfalls sieht die Ergebnistabelle wie die in der folgenden Abbildung aus.




Eine Kopfzeile gibt die Gesamtzahl der Datensätze an, die den Filterbedingungen entsprechen. Nach der Kopfzeile sehen Sie Folgendes.

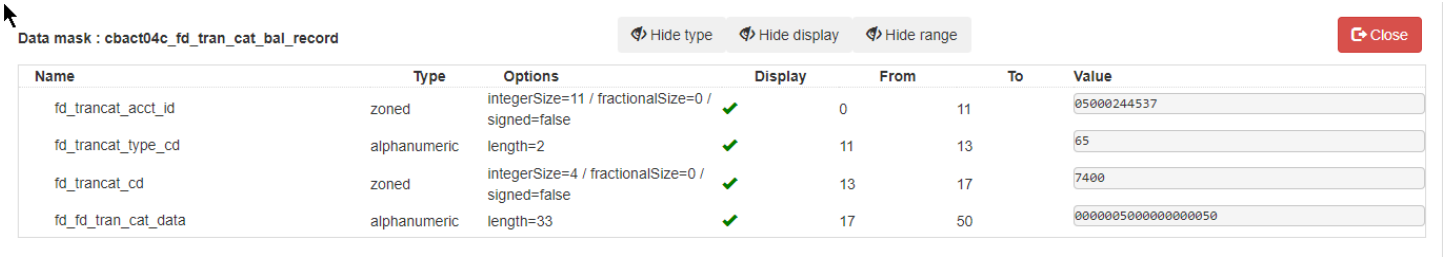
- Erinnerung an die verwendete Datenmaske (falls vorhanden) und die Filterbedingungen.
- Eine Aktualisierungsschaltfläche, mit der Sie die Aktualisierung der gesamten Ergebnistabelle mit den neuesten Werten aus dem Blusam-Speicher auslösen können (da sie beispielsweise von einem anderen Benutzer aktualisiert worden sein könnte).

Für jeden abgerufenen Datensatz enthält die Tabelle eine Zeile, in der das Ergebnis der Anwendung der Datenmaske auf den Inhalt der Datensätze angezeigt wird. Jede Spalte ist die Interpretation des Datensatzunterabschnitts entsprechend dem Spaltentyp (und unter Verwendung der ausgewählten Kodierung). Links neben jeder Zeile befinden sich drei Schaltflächen:

- eine Lupentaste: führt zu einer speziellen Seite, auf der der Inhalt des detaillierten Datensatzes angezeigt wird
- eine Stifttaste: führt zu einer speziellen Bearbeitungsseite für den Inhalt des Datensatzes:

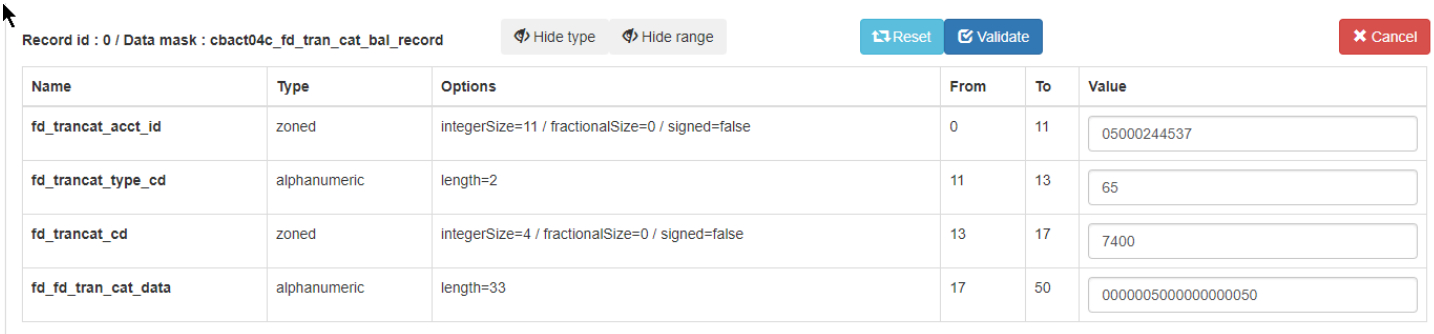
- eine Mülleimertaste: wird verwendet, um den angegebenen Datensatz aus dem Blusam-Speicher zu löschen

Den Inhalt des Datensatzes im Detail anzeigen:




- Drei Umschaltflächen zum Ein- oder Ausblenden einiger Spalten:
  - Typ ausblenden/einblenden
  - Display-Flagge ausblenden/einblenden
  - Bereich ausblenden/einblenden
- Um diese spezielle Seite zu verlassen und zur Ergebnistabelle zurückzukehren, wählen Sie Schließen.
- Jede Zeile steht für eine Spalte aus der Datenmaske mit den folgenden Spalten:
  - Name: Der Name der Spalte
  - Typ: Der Typ der Spalte
  - Anzeige: der Anzeigeindikator; ein grünes Häkchen wird angezeigt, wenn das passende Maskenelement mit definiert ist`skip = false`, andernfalls wird ein rotes Kreuz angezeigt
  - Von und bis: Der Bereich, der auf 0 basiert, für den Datensatzunterabschnitt
  - Wert: Der interpretierte Wert des Datensatzunterabschnitts unter Verwendung von Typ und Kodierung

Den Inhalt des Datensatzes bearbeiten:

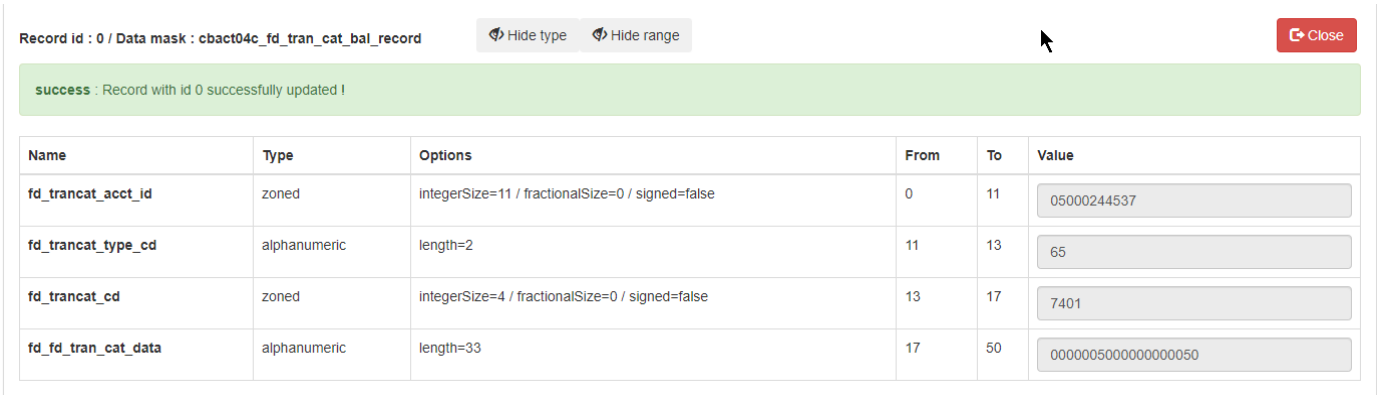


Die Bearbeitungsseite ähnelt der oben beschriebenen Ansichtsseite, mit der Ausnahme, dass die Werte der Maskenelemente bearbeitet werden können. Drei Schaltflächen steuern den Aktualisierungsvorgang:

- Reset: Setzt die editierbaren Werte auf die ursprünglichen Rekordwerte zurück (vor jeder Ausgabe);
- Validieren: validiert die Eingabe in Bezug auf den Maskenelementtyp. Für jedes Maskenelement wird das Ergebnis der Validierung mit visuellen Beschriftungen gedruckt (0Kund ein Kontrollkästchen, wenn die Validierung erfolgreich war, ERR0R und ein rotes Kreuz, wenn die Überprüfung fehlgeschlagen ist, sowie eine Fehlermeldung mit Hinweisen auf den Fehler bei der Überprüfung). Wenn die Validierung erfolgreich war, werden zwei neue Schaltflächen angezeigt:
  - Speichern: versucht, den vorhandenen Datensatz im Blusam-Speicher zu aktualisieren
  - Eine Kopie speichern: versucht, einen neuen Datensatz im Blusam-Speicher zu erstellen




- Wenn das Speichern des Datensatzes im Speicher erfolgreich ist, wird eine Meldung angezeigt und die Seite wechselt in einen schreibgeschützten Modus (Werte für Maskenelemente können nicht mehr bearbeitet werden):




- Wenn aus irgendeinem Grund die Persistenz des Datensatzes im Speicher fehlschlägt, wird eine rote Fehlermeldung mit Angabe der Fehlerursache angezeigt. Der häufigste Fall von Fehlern besteht darin, dass das Speichern des Datensatzes zu einer Beschädigung des Schlüssels

führen würde (ungültiger oder doppelter Schlüssel). Eine Veranschaulichung finden Sie in der folgenden Anmerkung.

  - Um den Vorgang zu beenden, wählen Sie die Schaltfläche „Schließen“.

- Abbrechen: Beendet die Bearbeitungssitzung, schließt die Seite und bringt Sie zurück zur Seite mit der Datensatzliste.

Hinweis:

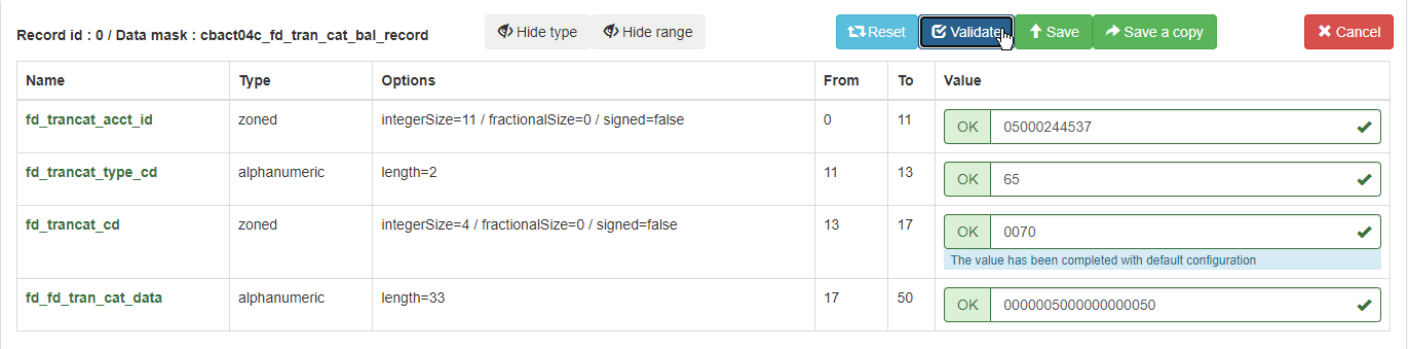
- Der Überprüfungsmechanismus überprüft nur, ob der Wert des Maskenelements formal mit dem Maskenelementtyp kompatibel ist. Sehen Sie sich zum Beispiel diese fehlgeschlagene Überprüfung für ein numerisches Maskenelement an:



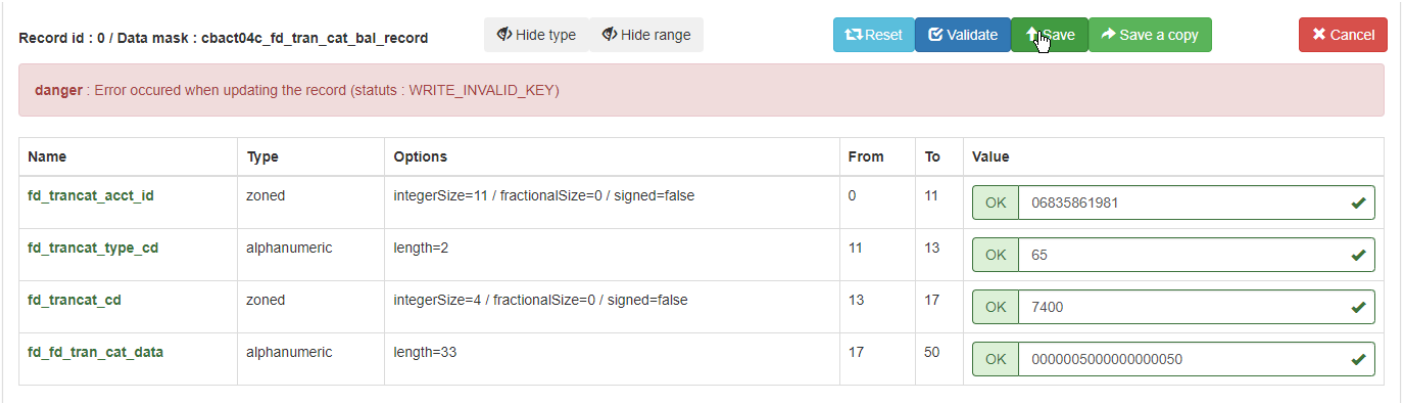

- Der Überprüfungsmechanismus versucht möglicherweise, ungültige Eingaben automatisch zu korrigieren, und zeigt je nach Typ eine blaue Informationsmeldung an, die darauf hinweist, dass der Wert automatisch korrigiert wurde. Geben Sie beispielsweise 7XX0 als numerischen Wert in das numerische Maskenelement ein: `fd_trncat_cd`




Der Aufruf der Validierung führt zu folgendem Ergebnis:

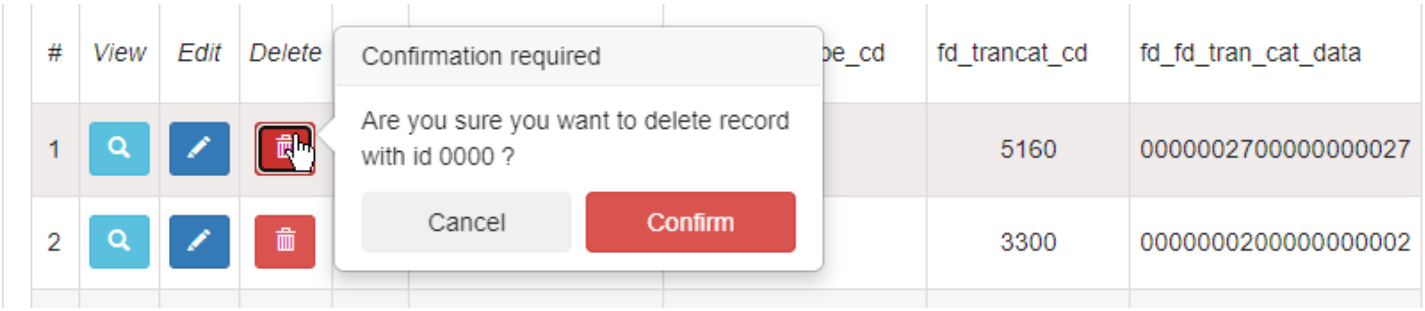



- Der Validierungsmechanismus überprüft nicht, ob der angegebene Wert im Hinblick auf die Schlüsselintegrität gültig ist (falls es sich um einen eindeutigen Schlüssel für den angegebenen Datensatz handelt). Wenn beispielsweise die Validierung erfolgreich war und die angegebenen Werte zu einer ungültigen oder doppelten Schlüsselsituation führen, schlägt die Persistenz fehl und es wird eine Fehlermeldung angezeigt:



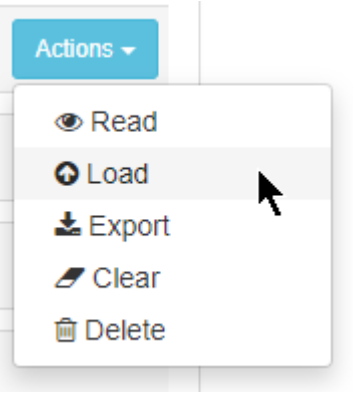

Löschen eines Datensatzes:

Um einen Datensatz zu löschen, wählen Sie die Papierkorbtaste:

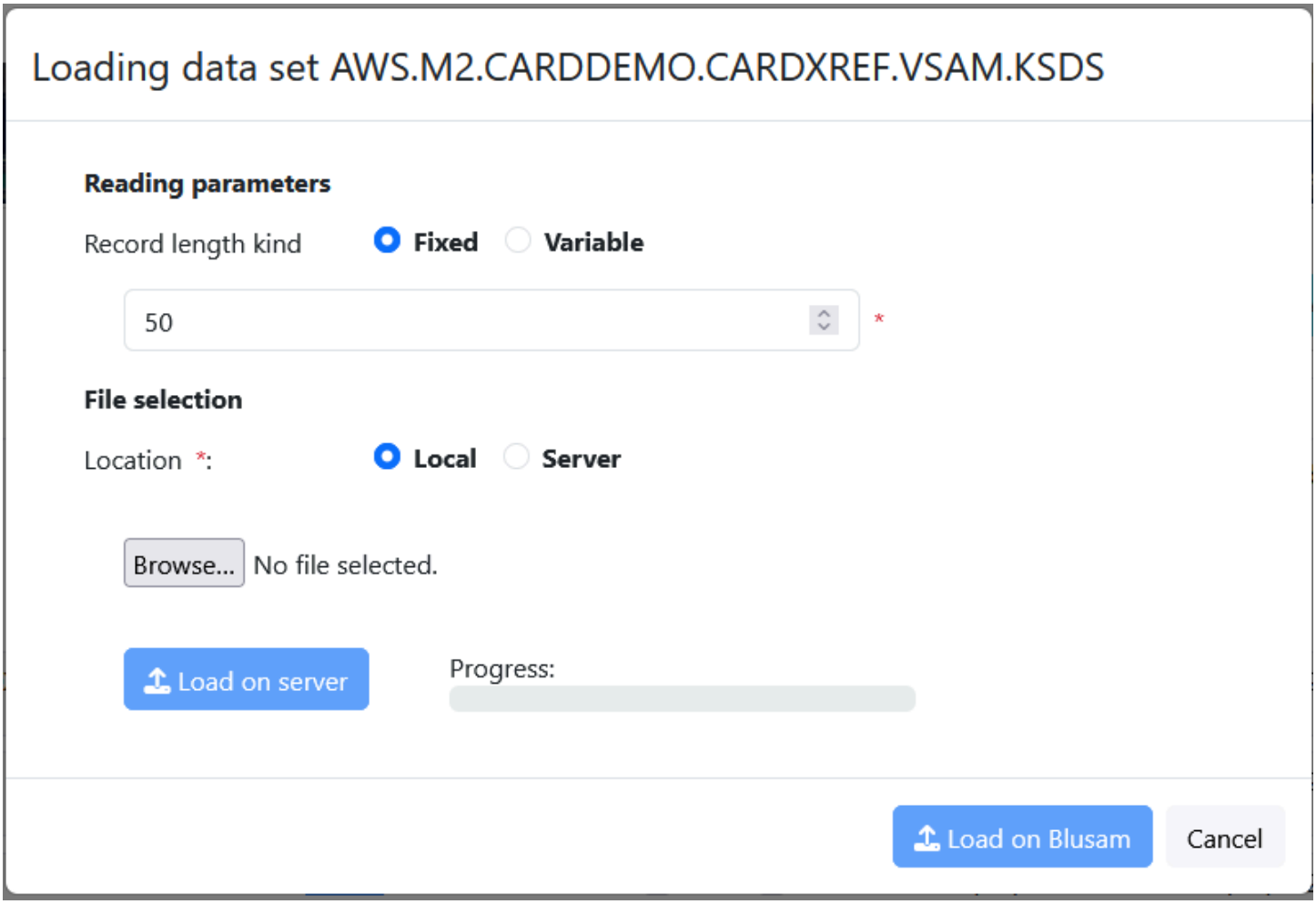



Datensätze in einen Datensatz laden

Um Datensätze in einen Datensatz zu laden, wählen Sie „Aktionen“ und anschließend „Laden“.




Ein Fenster mit Ladeoptionen wird angezeigt.

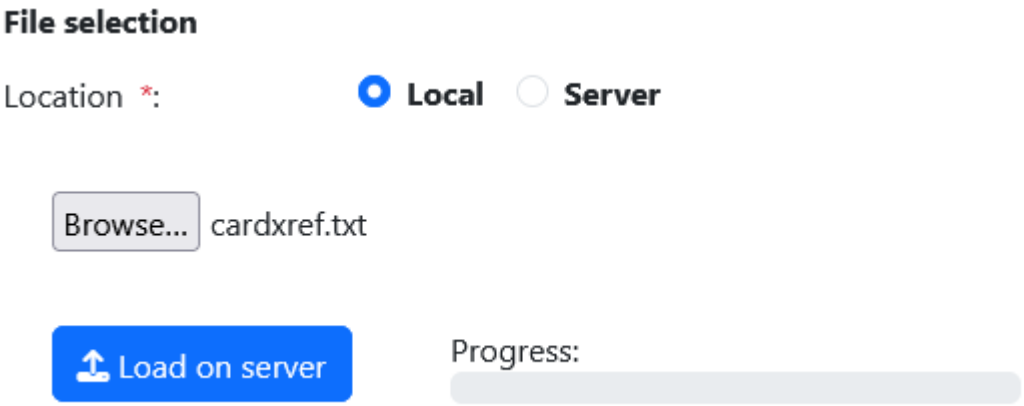



Zunächst sind sowohl die Schaltflächen Auf Server laden als auch Auf Blusam laden deaktiviert.

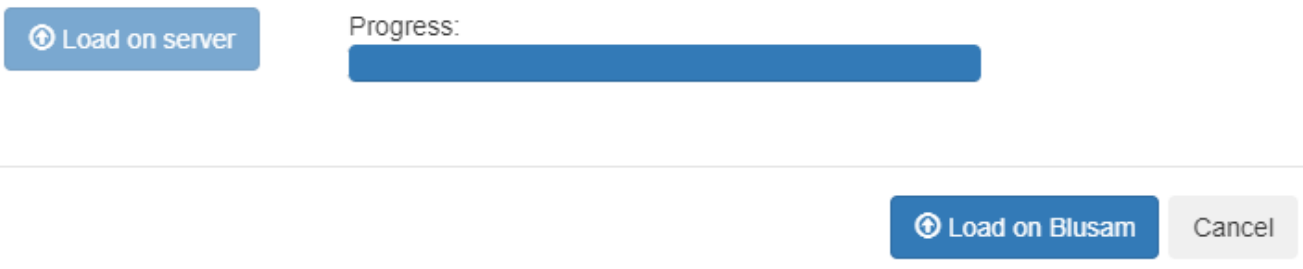
Parameter lesen:

- Art der Datensatzlänge:

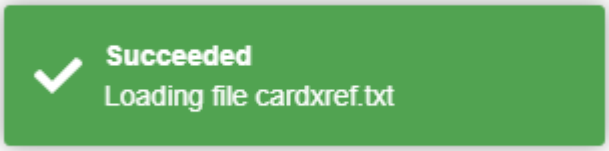
- Feste oder variable Datensatzlänge: Verwenden Sie das Optionsfeld, um anzugeben, ob beim Export älterer Datensätze Datensätze mit fester Länge oder Datensätze mit variabler Länge verwendet werden (die Datensätze werden voraussichtlich mit RDW-Bytes beginnen). Wenn Sie Fest wählen, muss die Datensatzlänge (in Byte) als positive Ganzzahl im Eingabefeld angegeben werden. Der Wert sollte vorab mit den Informationen aus dem Datensatz gefüllt werden. Wenn Sie Variable wählen, verschwindet das angegebene Eingabefeld.
- Dateiauswahl:
  - Lokal: Wählen Sie die Datensatzdatei von Ihrem lokalen Computer aus, indem Sie die Dateiauswahl unten verwenden (Hinweis: Die Dateiauswahl verwendet das Gebietsschema Ihres Browsers zum Drucken der Nachrichten -- hier auf Französisch, aber es könnte auf Ihrer Seite anders aussehen, was zu erwarten ist). Nachdem Sie die Auswahl getroffen haben, wird das Fenster mit dem Namen der Datendatei aktualisiert und die Schaltfläche Auf Server laden ist aktiviert:



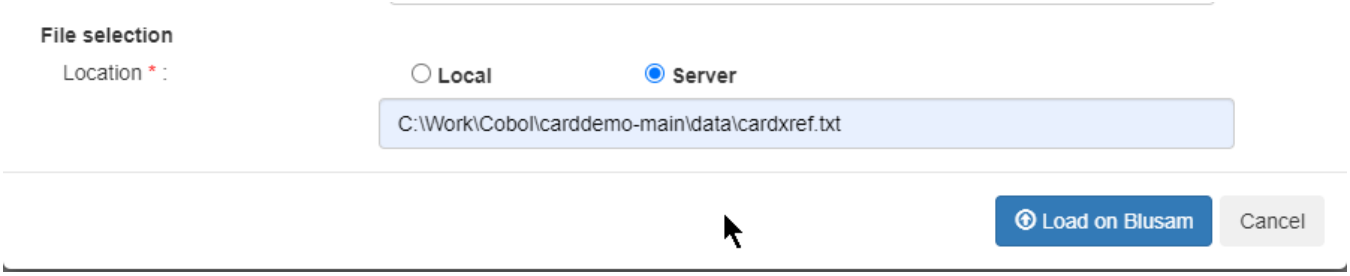

Wählen Sie Auf Server laden. Wenn der Fortschrittsbalken sein Ende erreicht hat, wird die Schaltfläche Load on Blusam aktiviert:



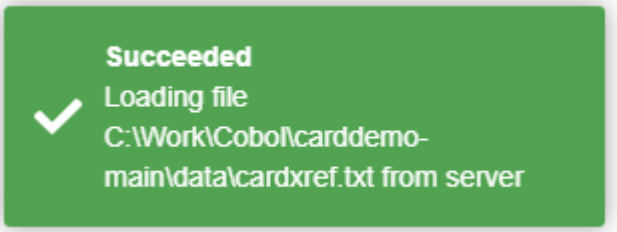

Um den Ladevorgang in den Blusam-Speicher abzuschließen, wählen Sie Load on Blusam. Wählen Sie andernfalls Abbrechen. Wenn Sie den Ladevorgang fortsetzen möchten, erscheint nach Abschluss des Ladevorgangs in der unteren rechten Ecke eine Benachrichtigung:




- Server: Wenn Sie diese Option wählen, erscheint ein Eingabefeld, während die Schaltfläche Auf Server laden ausgeblendet wird. Im Eingabefeld müssen Sie den Pfad zur Datensatzdatei auf dem Blusam-Server angeben (dies setzt voraus, dass Sie die angegebene Datei zuerst auf den Blusam-Server übertragen haben). Nachdem Sie den Pfad angegeben haben, wird Load on Blusam aktiviert:

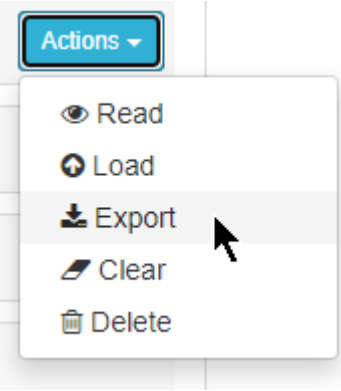



Um den Ladevorgang abzuschließen, wählen Sie Load on Blusam. Wählen Sie andernfalls Abbrechen. Wenn Sie mit dem Laden fortfahren möchten, erscheint nach Abschluss des Ladevorgangs eine Benachrichtigung. Die Benachrichtigung unterscheidet sich vom Ladevorgang aus dem Browser, da sie den Pfad des Datendateiservers gefolgt von den Worten vom Server anzeigt:

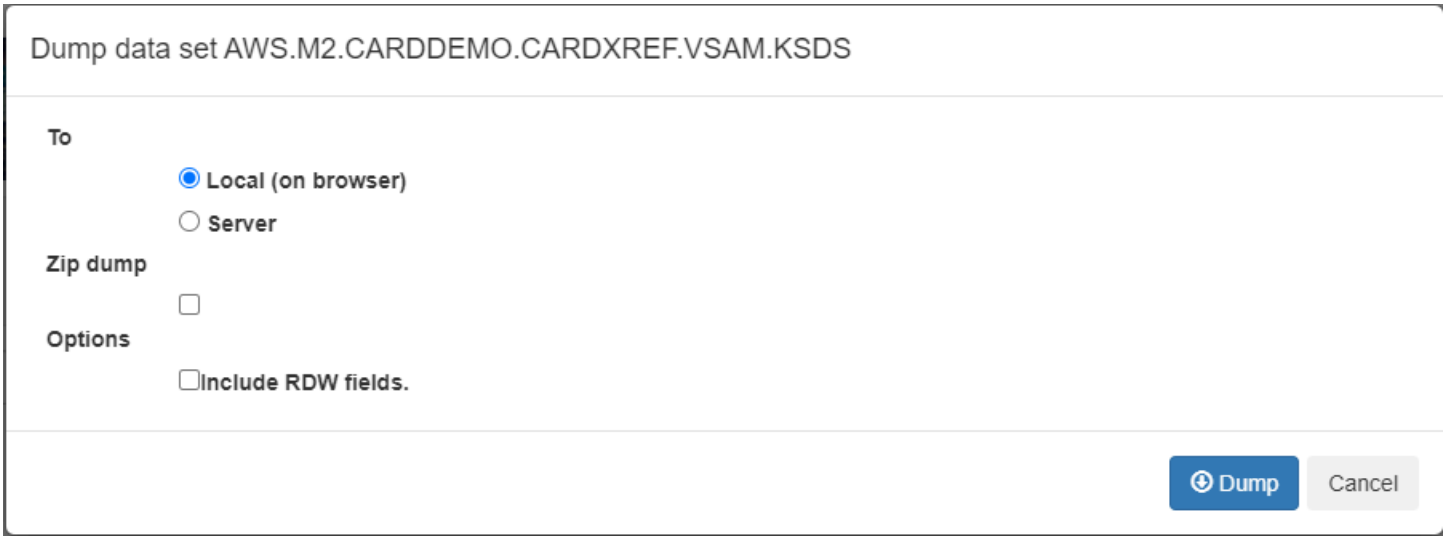



Datensätze aus einem Datensatz exportieren

Um Datensatzdatensätze zu exportieren, wählen Sie in der aktuellen Datensatzzeile Aktionen und dann Exportieren aus:




Das folgende Popup-Fenster wird angezeigt.




Optionen:

An: eine Optionsschaltfläche, um das Exportziel auszuwählen, entweder als Download im Browser (Lokal (im Browser)) oder in einen bestimmten Ordner auf dem Server, auf dem die BAC-Anwendung gehostet wird. Wenn Sie sich für den Export mit der Serverauswahl entscheiden, wird ein neues Eingabefeld angezeigt:



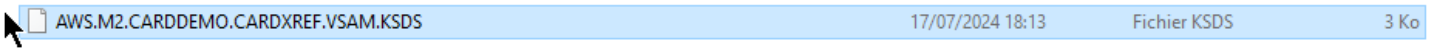

Wie das rote Sternchen rechts neben dem Eingabefeld anzeigt, ist es erforderlich, einen gültigen Ordnerpfad auf dem Server anzugeben (die Schaltfläche Dump ist inaktiv, solange kein Ordnerspeicherort angegeben wurde).

Um auf den Server zu exportieren, müssen Sie über die erforderlichen Zugriffsrechte für das Server-Dateisystem verfügen, falls Sie die exportierte Datensatzdatei nach dem Export bearbeiten möchten.

Zip-Dump: Ein Kontrollkästchen, das ein komprimiertes Archiv anstelle einer Rohdatei erzeugt.

Optionen: Um bei Datensätzen mit variabler Länge am Anfang jedes Datensatzes im exportierten Datensatz ein Record Descriptor Word (RDW) einzufügen, wählen Sie „RDW-Felder einbeziehen“.

Um den Datensatz-Exportvorgang zu starten, wählen Sie Dump. Wenn Sie in den Browser exportieren möchten, überprüfen Sie den Download-Ordner für die Exportdatensatzdatei. Die Datei wird denselben Namen wie der Datensatz haben:

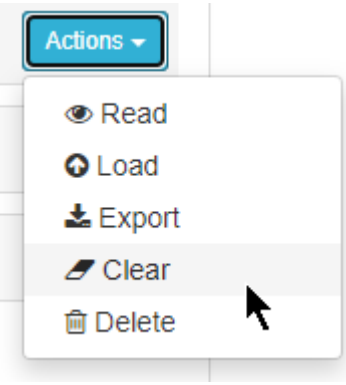



Hinweis:

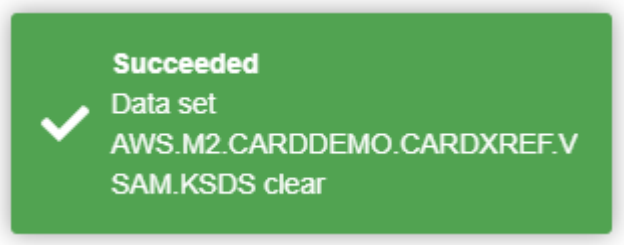
- Für KSDS werden die Datensätze in der Reihenfolge der Primärschlüssel exportiert.
- Für ESDS und RRDS werden die Datensätze in der Reihenfolge RBA (Relative Byte Address) exportiert.
- Für alle Arten von Datensätzen werden die Datensätze als rohe binäre Arrays exportiert (es findet keinerlei Konvertierung statt), wodurch die direkte Kompatibilität mit älteren Plattformen gewährleistet ist.

Löschen von Datensätzen aus einem Datensatz

Um alle Datensätze aus einem Datensatz zu löschen, wählen Sie „Aktionen“ und anschließend „Löschen“:

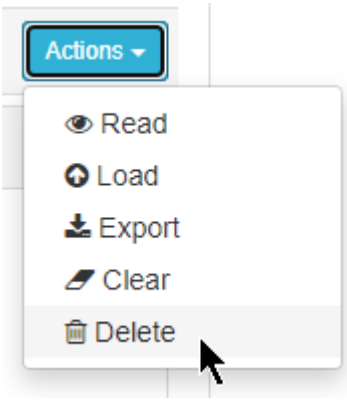



Nachdem alle Datensätze aus einem Datensatz entfernt wurden, wird die folgende Benachrichtigung angezeigt.

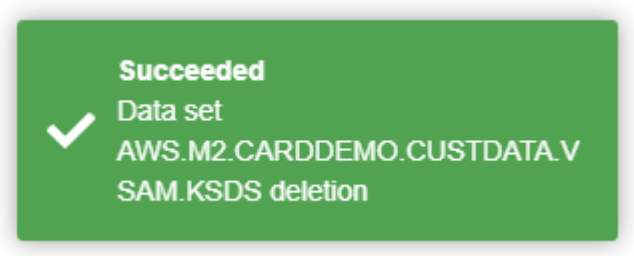



Löschen eines Datensatzes

Um einen Datensatz zu löschen, wählen Sie „Aktionen“ und anschließend „Löschen“:




Nachdem Sie einen Datensatz gelöscht haben, wird die folgende Benachrichtigung angezeigt:



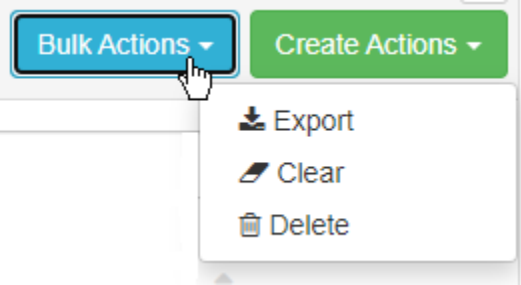

Massenoperationen

Drei Massenoperationen sind für Datensätze verfügbar:

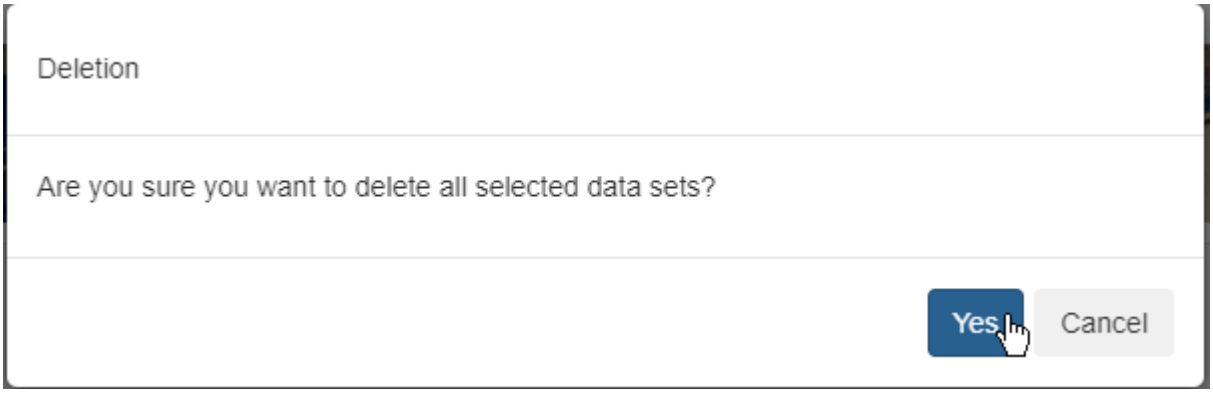
- Export
- Löschen Sie
- Löschen

Massenoperationen können nur auf eine Auswahl von Datensätzen angewendet werden (mindestens ein Datensatz muss ausgewählt werden). Die Auswahl von Datensätzen erfolgt durch Ankreuzen der

Auswahlkästchen links neben den Datensatzzeilen in der Tabelle mit den Datensätzen. Wenn Sie mindestens einen Datensatz auswählen, wird die Dropdownliste Massenaktionen aktiviert:



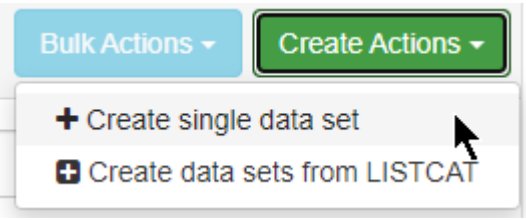

Abgesehen von der Tatsache, dass die angegebenen Aktionen für eine Auswahl von Datensätzen und nicht für einen einzelnen Datensatz gelten, ähneln die Aktionen den oben beschriebenen. Einzelheiten finden Sie in der Dokumentation zu den entsprechenden Aktionen. Der Textinhalt des Popup-Fensters wird leicht unterschiedlich sein, um den Bulk-Charakter widerzuspiegeln. Wenn Sie beispielsweise versuchen, mehrere Datensätze zu löschen, sieht das Popup-Fenster wie folgt aus:



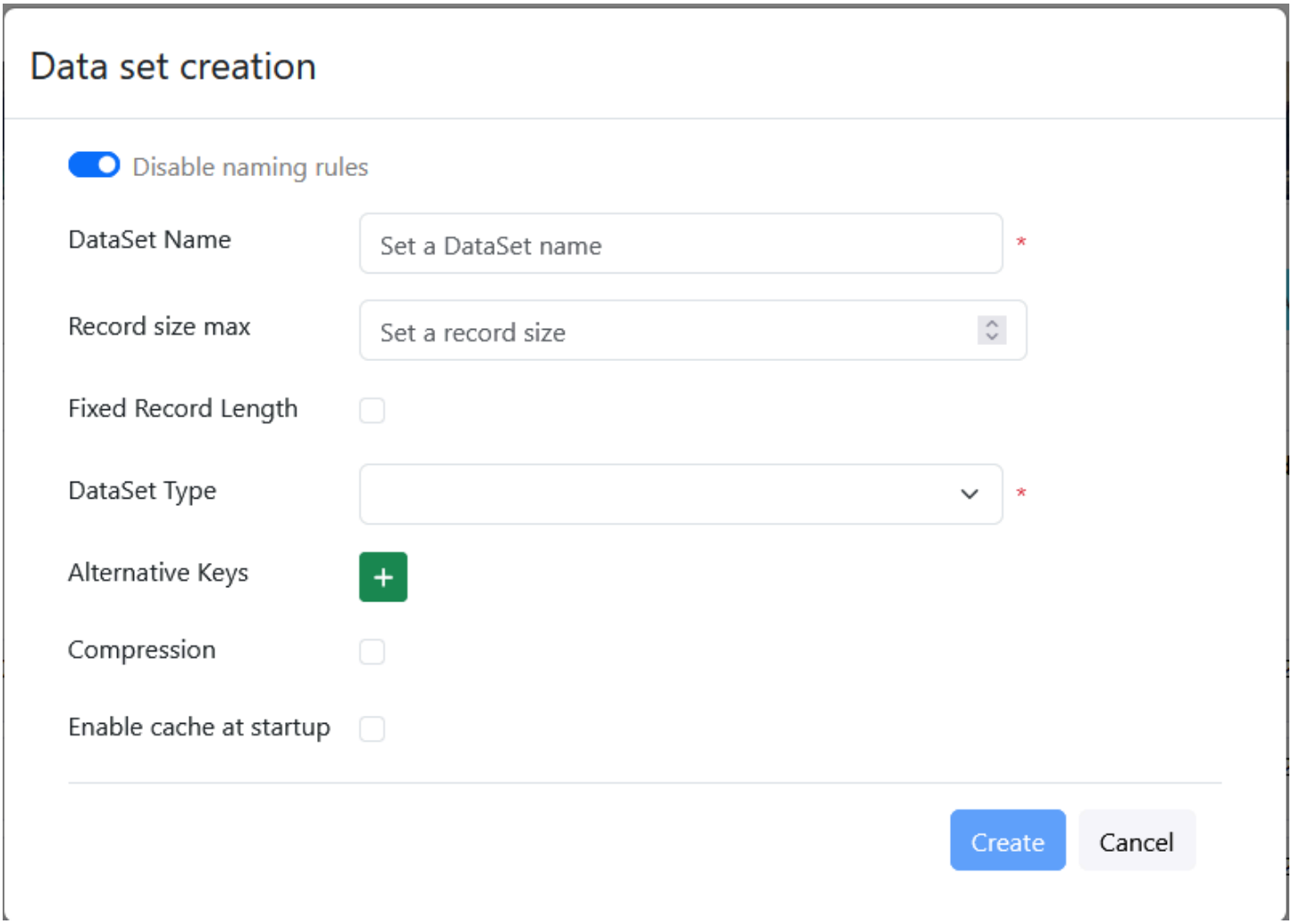

Operationen erstellen

Erstellen Sie einen einzelnen Datensatz

Wählen Sie „Aktionen“ und anschließend „Einzelner Datensatz erstellen“:

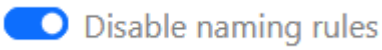


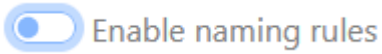

Das Formular zur Erstellung eines Datensatzes wird dann als Popup-Fenster angezeigt:



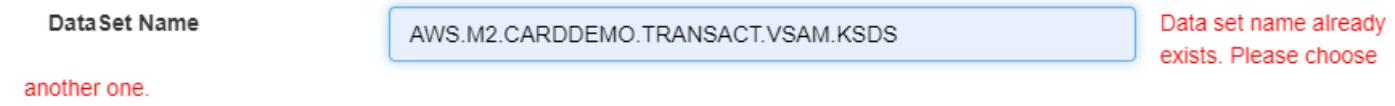

Sie können die folgenden Attribute für die Datensatzdefinition angeben:

- Benennungsregeln aktivieren und deaktivieren: Verwenden Sie das Umschalt-Widget „Benennungsregeln deaktivieren/Benennungsregeln aktivieren“, um Benennungskonventionen für Datensätze zu deaktivieren und zu aktivieren. Wir empfehlen, dass Sie den Schalter auf dem Standardwert belassen und die Benennungsregeln für Datensätze aktiviert sind (das Umschalt-Widget sollte „Benennungsregeln deaktivieren“ anzeigen):

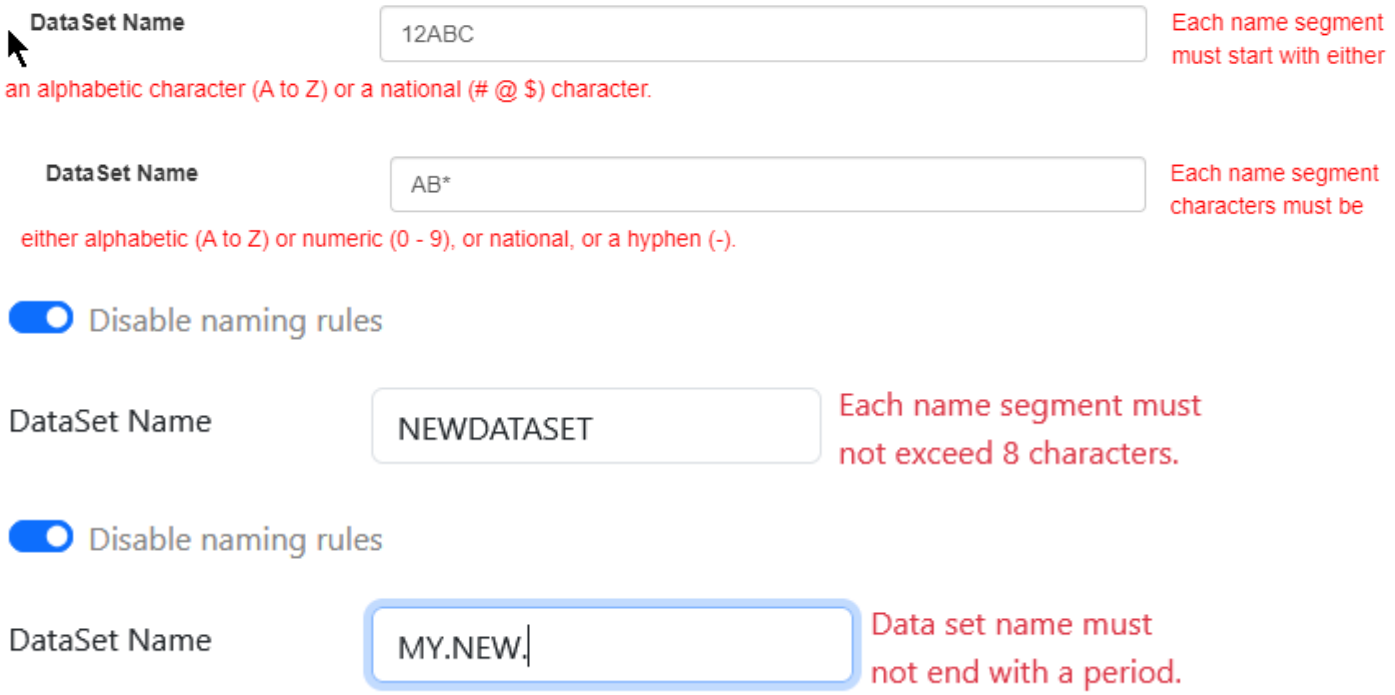
  
  

- Name des Datensatzes: Der Name für den Datensatz. Wenn Sie einen Namen angeben, der bereits verwendet wird, wird die folgende Fehlermeldung angezeigt.



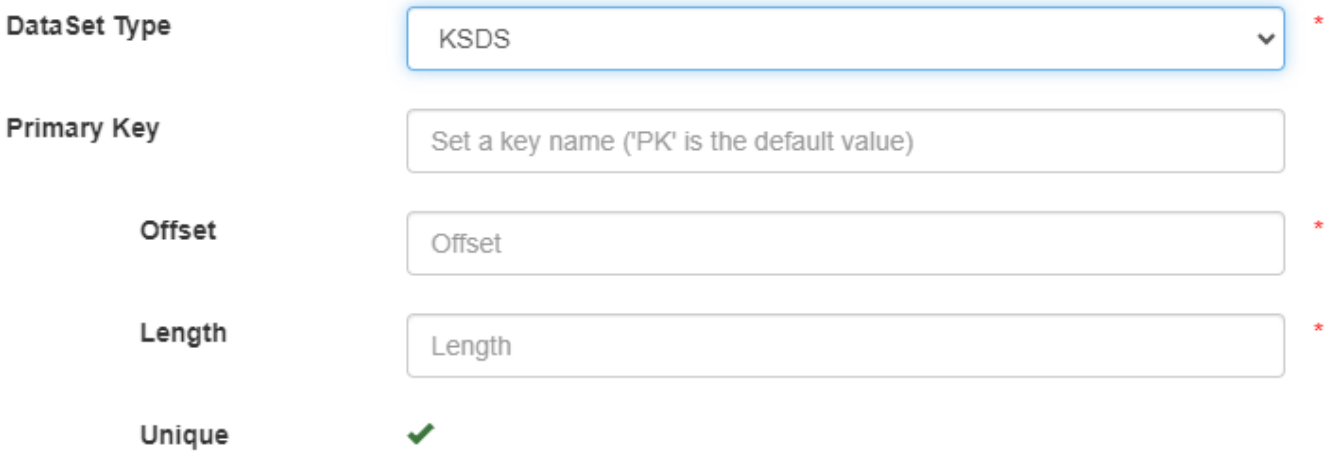

Der Name muss außerdem die Benennungskonvention einhalten, falls er aktiviert ist:




- Max. Datensatzgröße: Dies muss eine positive Ganzzahl sein, die die Datensatzgröße für einen Datensatz mit Datensätzen fester Länge darstellt. Bei Datensätzen mit Datensätzen variabler Länge können Sie das Feld leer lassen.
- Datensatz mit fester Länge: Ein Kontrollkästchen, um anzugeben, ob die Datensatzlänge fest oder variabel ist. Wenn diese Option ausgewählt ist, enthält der Datensatz Datensätze mit fester Länge, andernfalls ist die Datensatzlänge variabel.

  Wenn Sie ältere Daten in einen Datensatz mit variabler Länge importieren, müssen die bereitgestellten älteren Datensätze das Record Descriptor Word (RDW) enthalten, das die Länge jedes Datensatzes angibt.
- Datensatztyp: Eine Dropdownliste zur Angabe des aktuellen Datensatztyps. Die folgenden Typen werden unterstützt.
  - ESDS
  - Große ESDs
  - KINDER

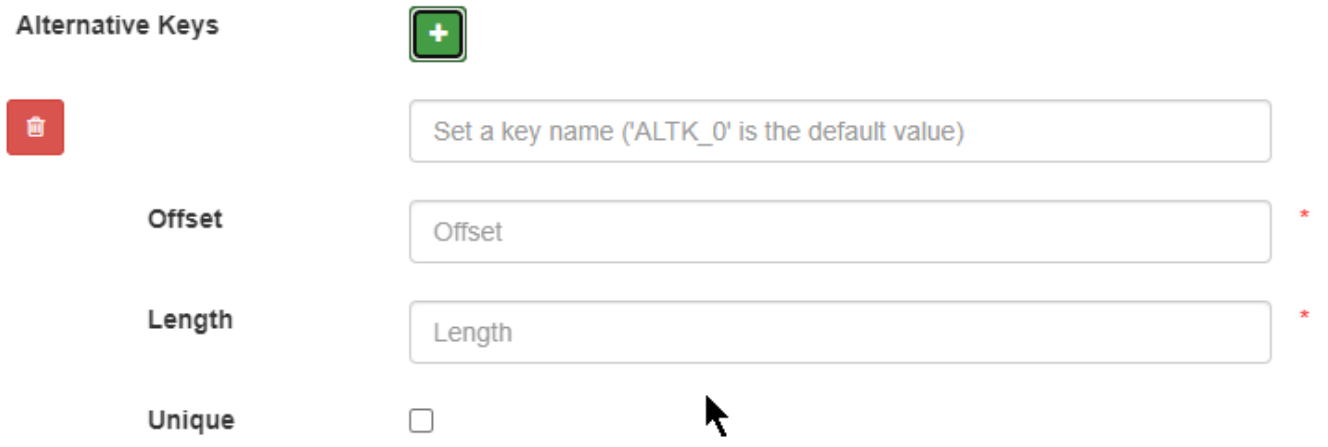
Für KSDS müssen Sie den Primärschlüssel angeben:




Geben Sie für den Primärschlüssel Folgendes an:

- Name: Dieses Feld ist optional. Der Standardwert ist **PK**.
- Offset: Der 0-basierte Offset des Primärschlüssels innerhalb des Datensatzes. Der Offset muss eine positive Ganzzahl sein. Dies ist ein Pflichtfeld.
- Länge: Die Länge des Primärschlüssels. Diese Länge muss eine positive Ganzzahl sein. Dies ist ein Pflichtfeld.

Für KSDS und ESDS können Sie optional eine Sammlung von Alternativschlüsseln definieren, indem Sie auf die Plus-Schaltfläche vor der Bezeichnung Alternative Keys klicken. Jedes Mal, wenn Sie auf diese Schaltfläche klicken, wird im Formular zur Datensatzerstellung ein neuer Abschnitt zur Definition alternativer Schlüssel angezeigt:



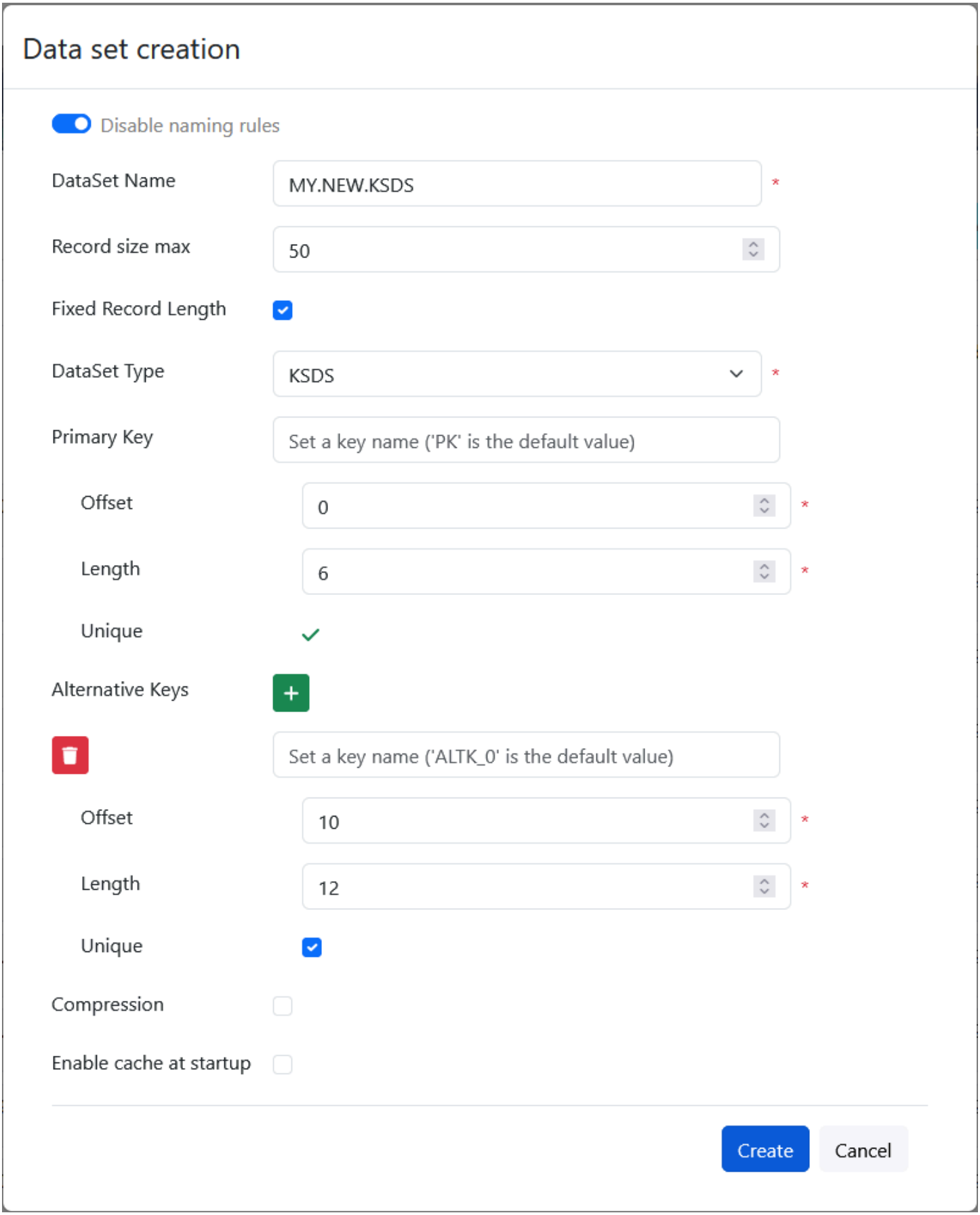
Für jeden alternativen Schlüssel müssen Sie Folgendes angeben:

- Name: Dieses Feld ist optional. Der Standardwert ist**`ALTK_#`**, wobei **#** für einen automatisch inkrementierten Zähler steht, der bei 0 beginnt.
- Offset: Der 0-basierte Offset des alternativen Schlüssels innerhalb des Datensatzes. Muss eine positive Ganzzahl sein. Dies ist ein Pflichtfeld.
- Länge: Die Länge des alternativen Schlüssels. Diese Länge muss eine positive Ganzzahl sein. Dies ist ein Pflichtfeld.
- Eindeutig: Ein Kontrollkästchen, das angibt, ob der alternative Schlüssel doppelte Eingaben akzeptiert. Wenn diese Option ausgewählt ist, wird der alternative Schlüssel als einzigartig definiert (er akzeptiert KEINE doppelten Schlüsseleingaben). Dies ist ein Pflichtfeld.

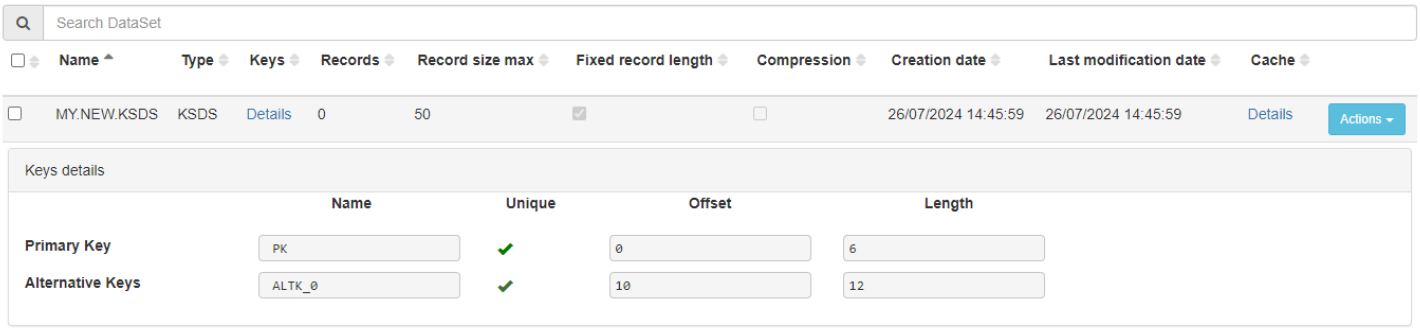
Um die alternative Schlüsseldefinition zu entfernen, verwenden Sie die Papierkorbschaltfläche auf der linken Seite.

- Komprimierung: Ein Kontrollkästchen, um anzugeben, ob die Komprimierung zum Speichern des Datensatzes verwendet werden soll.
- Cache beim Start aktivieren: Ein Kontrollkästchen, um anzugeben, ob der Datensatz beim Start der Anwendung in den Cache geladen werden soll.

Nachdem Sie die Attributdefinitionen angegeben haben, wählen Sie Erstellen, um fortzufahren:


Data set creation
Disable naming rules
DataSet Name
MY.NEW.KSDS
Record size max
50
Fixed Record Length
DataSet Type
KSDS
Primary Key
Set a key name ('PK' is the default value)
Offset
0
Length
6
Unique
Alternative Keys
Set a key name ('ALTK_0' is the default value)
Offset
10
Length
12
Unique
Compression
Enable cache at startup
Create
Cancel

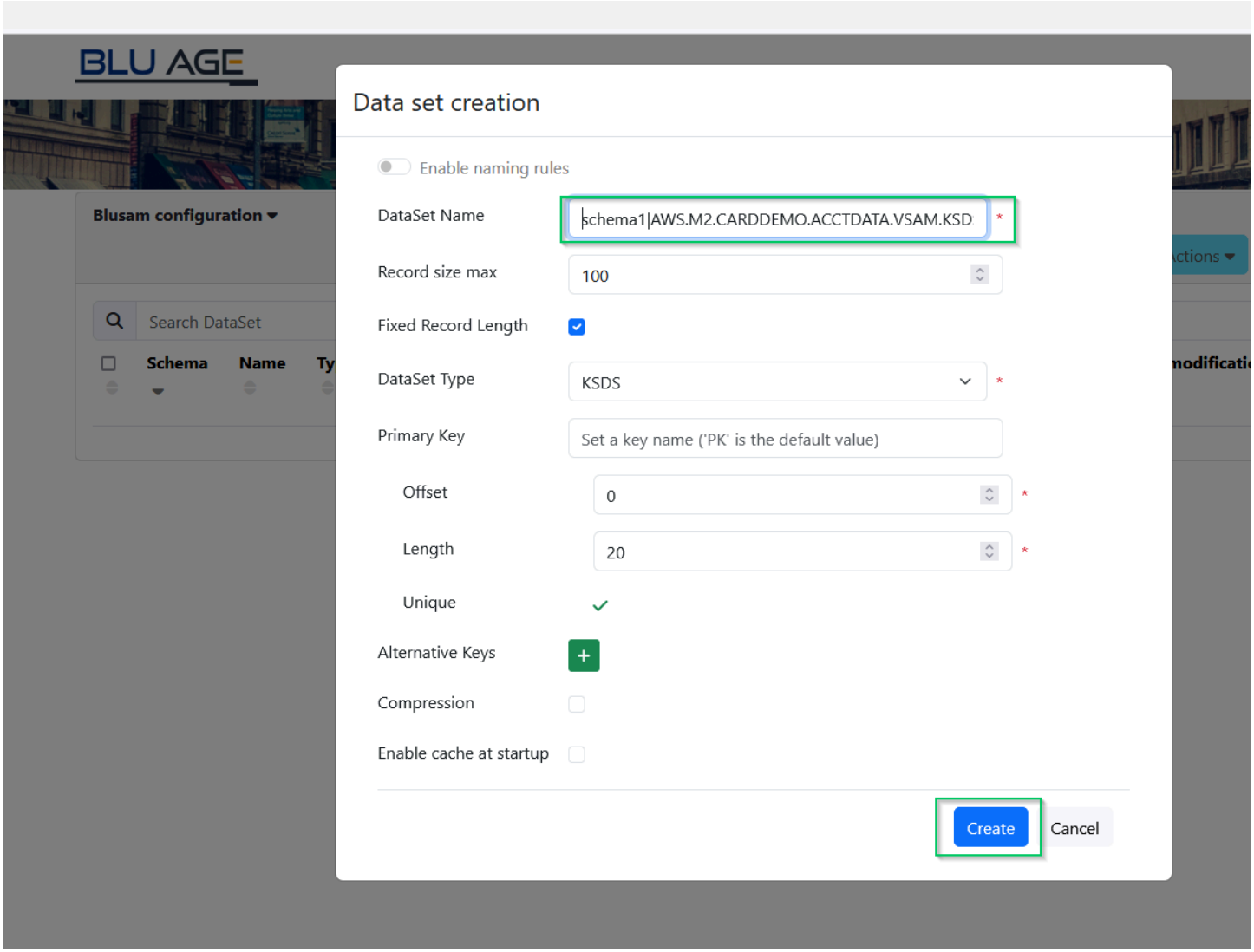
Das Erstellungsfenster wird geschlossen und die Startseite mit der Liste der Datensätze wird angezeigt. Sie können die Details des neu erstellten Datensatzes einsehen.




Erstellen Sie einen einzelnen Datensatz im Multischemamodus

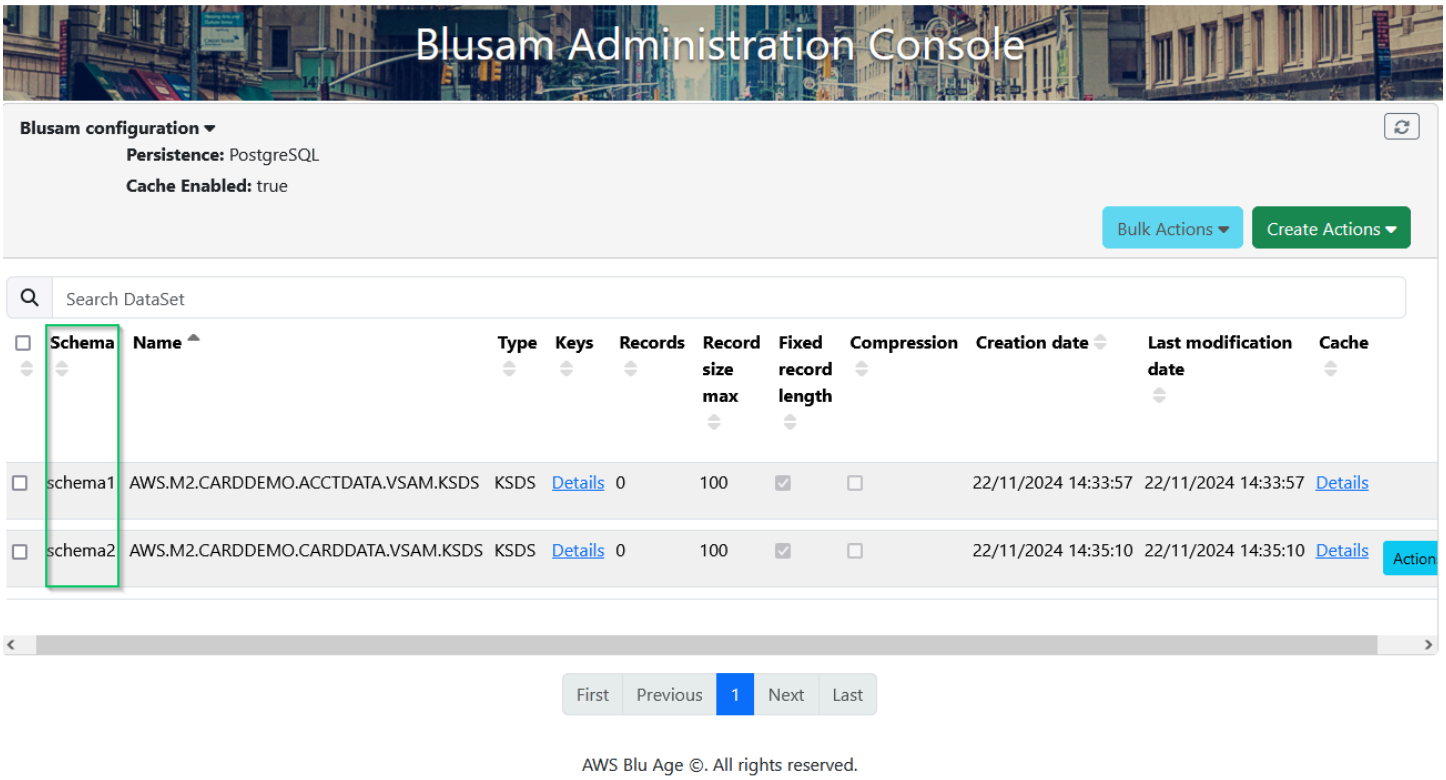
Ein Datensatz kann in einem Multischemamodus erstellt werden, indem dem Namen des Datensatzes der Schemaname gefolgt von einem senkrechten Strich (|) vorangestellt wird (z. B.). `schema1|AWS.M2.CARDDEMO.ACCTDATA.VSAM.KSDS`

> **Note**
>
> Das für die Erstellung des Datensatzes verwendete Schema sollte in der `application-main.yml` Konfiguration angegeben werden. Weitere Informationen finden Sie unter the section called “Eigenschaften der Konfiguration mehrerer Schemas ”.




Wenn kein Schemapräfix angegeben wird, wird der Datensatz in dem Standardschema erstellt, das in der Blusam-Datenquellen-URL in der Blusam-Datenquellen-Konfiguration angegeben ist. Wenn in der Blusam-Datenquellen-URL kein Schema angegeben ist, wird standardmäßig das Schema „public“ verwendet.

> **Note**
>
> Im Multischemamodus zeigt die BAC-Konsole die Schemainformationen des Datensatzes in der ersten Spalte an.




Erstellen Sie Datensätze aus LISTCAT

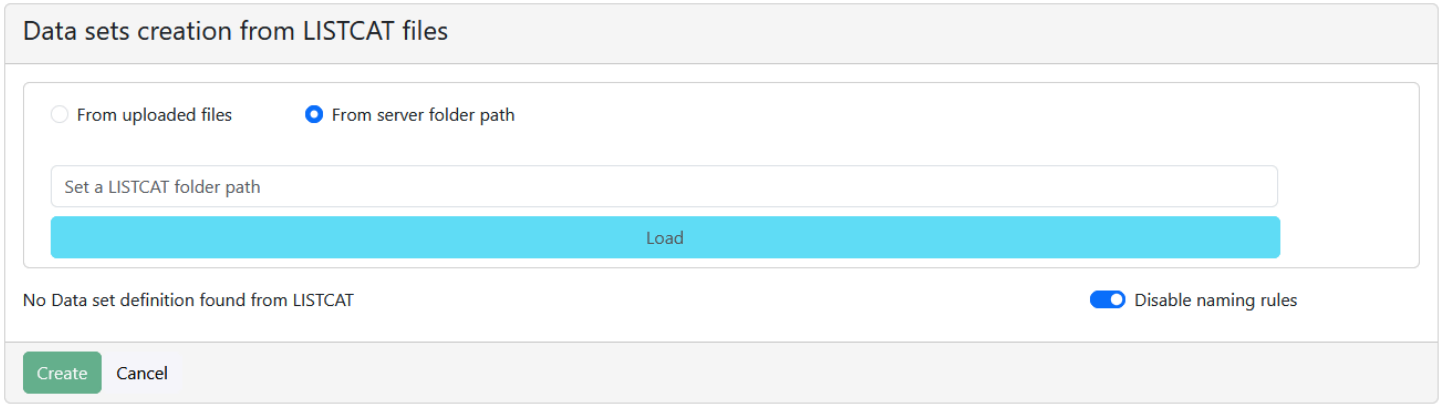
Diese Funktion ermöglicht es, die Vorteile der LISTCAT-JSON-Dateien zu nutzen, die während des BluAge Transformationsprozesses mit BluInsights Transformation Center als Ergebnis der Analyse des LISTCAT-Exports aus den älteren Plattformen erstellt wurden: LISTCAT-Exporte werden analysiert und in JSON-Dateien umgewandelt, die die Datensatzdefinitionen enthalten (Namen, Datensatztyp, Schlüsseldefinitionen und ob die Datensatzlänge fest oder variabel ist).

Mit den LISTCAT-JSON-Dateien können Datensätze direkt erstellt werden, ohne dass alle für Datensätze erforderlichen Informationen manuell eingegeben werden müssen. Sie können auch direkt eine Sammlung von Datensätzen erstellen, anstatt sie einzeln erstellen zu müssen.

Wenn für Ihr Projekt keine LISTCAT-JON-Datei verfügbar ist (z. B. weil zum Zeitpunkt der Transformation keine LISTCAT-Exportdatei verfügbar war), können Sie jederzeit manuell eine erstellen, vorausgesetzt, Sie halten sich an das im Anhang beschriebene LISTCAT-JSON-Format.

Wählen Sie in der Dropdownliste Aktionen erstellen die Option Datensätze aus LISTCAT erstellen aus.

Die folgende spezielle Seite wird angezeigt:



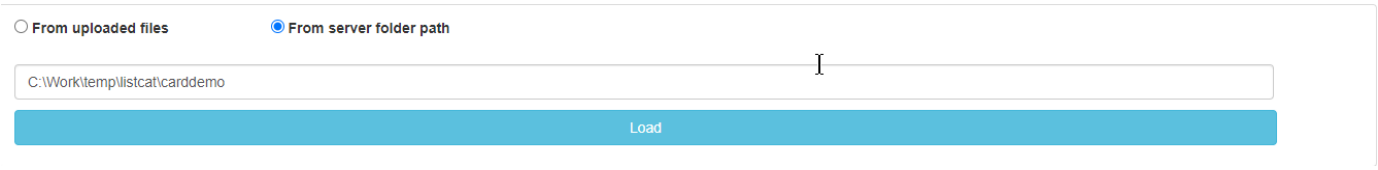

Zu diesem Zeitpunkt ist die Schaltfläche Laden deaktiviert, was zu erwarten ist.

Verwenden Sie die Optionsfelder, um anzugeben, wie Sie die LISTCAT-JSON-Dateien bereitstellen möchten. Es gibt zwei Optionen:

- Sie können Ihren Browser verwenden, um die JSON-Dateien hochzuladen.
- Sie können die JSON-Dateien von einem Ordnerspeicherort auf dem Server auswählen. Um diese Option zu wählen, müssen Sie zuerst die JSON-Dateien mit den entsprechenden Zugriffsrechten in den angegebenen Ordnerpfad auf dem Server kopieren.

Um JSON-Dateien auf dem Server zu verwenden

1. Legen Sie den Ordnerpfad auf dem Server fest und zeigen Sie auf den Ordner, der die LISTCAT-JSON-Dateien enthält:



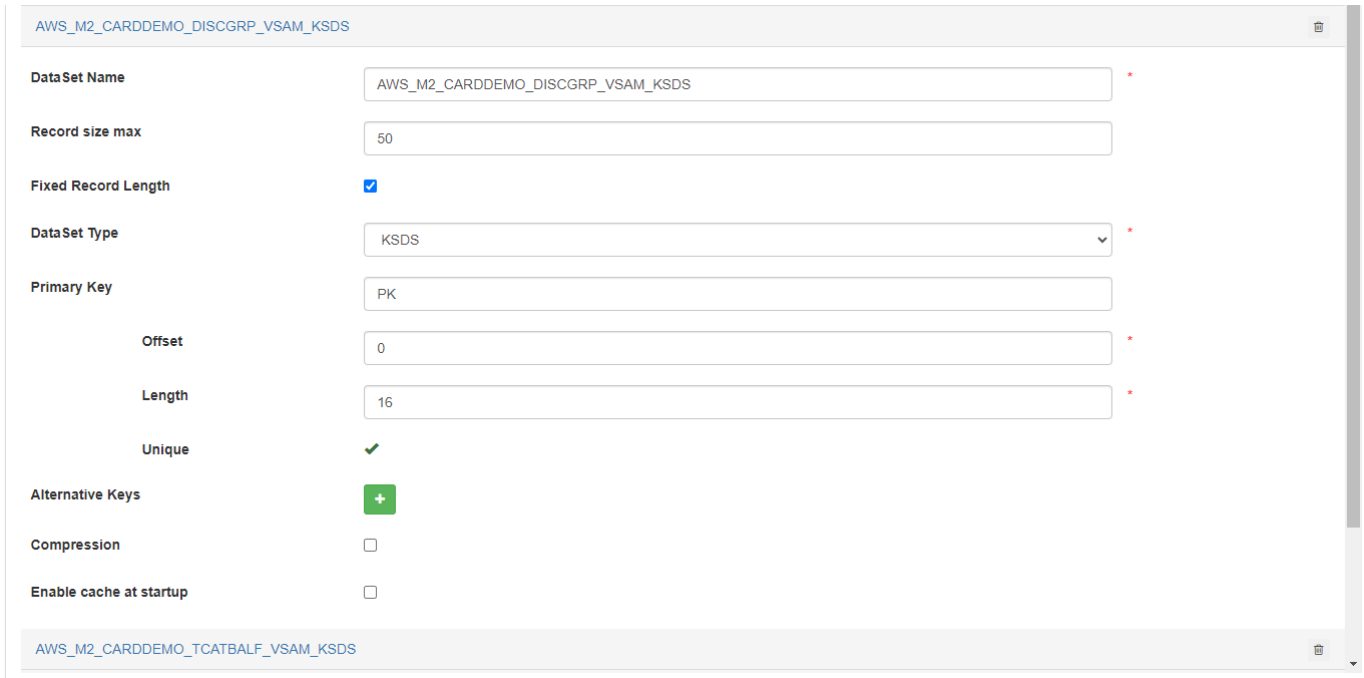

2. Wählen Sie die Schaltfläche Laden. Alle erkannten Datensatzdefinitionen werden in einer Tabelle aufgelistet:



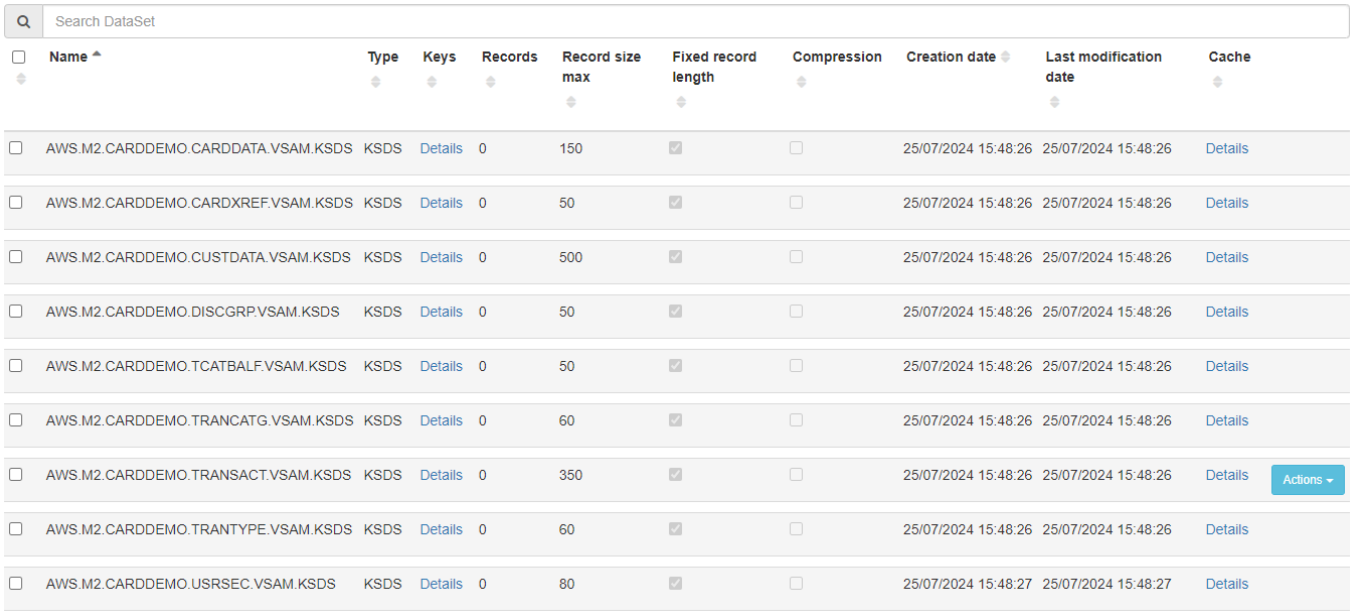

Jede Zeile steht für eine Datensatzdefinition. Sie können die Papierkorbschaltfläche verwenden, um eine Datensatzdefinition aus der Liste zu entfernen.

> **Important**
>
> Die Entfernung aus der Liste erfolgt sofort, ohne dass eine Warnmeldung angezeigt wird.

3. Der Name auf der linken Seite ist ein Link. Sie können ihn auswählen, um die Details der Datensatzdefinition, die bearbeitet werden kann, ein- oder auszublenden. Sie können die Definition frei ändern, wobei Sie auf der Grundlage der analysierten JSON-Datei beginnen.

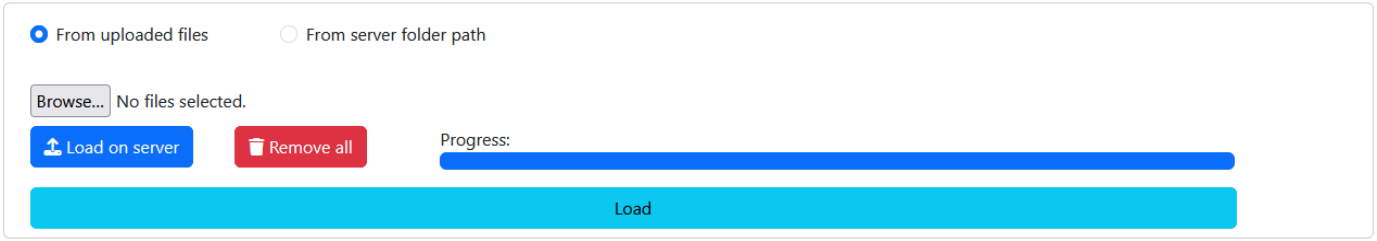


4. Um alle Datensätze zu erstellen, wählen Sie Create. Alle Datensätze werden erstellt und auf der Ergebnisseite der Datensätze angezeigt. Die neu erstellten Datensätze werden alle 0 Datensätze haben.




Um Dateien auf den Server hochzuladen

1. Diese Option ähnelt der Verwendung der Dateien aus dem Serverordnerpfad. In diesem Fall müssen Sie die Dateien jedoch zuerst mithilfe der Dateiauswahl hochladen. Wählen Sie alle Dateien aus, die Sie von Ihrem lokalen Computer hochladen möchten, und wählen Sie dann Auf Server laden.




2. Wenn der Fortschrittsbalken das Ende erreicht, wurden alle Dateien erfolgreich auf den Server hochgeladen und die Schaltfläche Laden ist aktiviert. Wählen Sie die Schaltfläche Laden und verwenden Sie die erkannten Datensatzdefinitionen wie zuvor erklärt.

## LISTCAT-JSON-Format

Das LISTCAT-JSON-Format wird durch die folgenden Attribute definiert:

- optional „catalogId“: Kennung des Legacy-Katalogs als Zeichenfolge oder „Standard“ für den Standardkatalog.
- „Identifier“: der Name des Datensatzes als Zeichenfolge.
- „isIndexed“: ein boolesches Flag zur Kennzeichnung von KSDS: true für KSDS, andernfalls false.
- „isLinear“: ein boolesches Flag zur Kennzeichnung von ESDS: true für ESDS, andernfalls false.
- „isRelative“: ein boolesches Flag zur Kennzeichnung von RRDS: true für RRDS, andernfalls false
- Hinweis: „isIndexed“, „isLinear“ und „isRelative“ schließen sich gegenseitig aus.
- „isFixedLengthRecord“: ein boolesches Flag: wird auf „true“ gesetzt, wenn Datensätze mit fester Länge gespeichert sind, andernfalls „false“.
- "avgRecordSize„: Durchschnittliche Datensatzgröße in Byte, ausgedrückt als positive Ganzzahl.
- "maxRecordSize„: Maximale Datensatzgröße in Byte, ausgedrückt als Ganzzahl. Sollte avgRecordSize für Datensatzgröße mit fester Länge gleich sein.
- Nur für KSDS: Obligatorische Primärschlüsseldefinition (als verschachteltes Objekt)
  - beschriftet mit „PrimaryKey“
  - „Offset“: 0-basierter Byte-Offset für den Primärschlüssel im Datensatz.
  - „Länge“: Länge des Primärschlüssels in Byte.
  - „unique“: muss für den Primärschlüssel auf true gesetzt werden.
- für KSDS/ESDS, Sammlung von Alternativschlüsseln (als Sammlung verschachtelter Objekte):
  - beschriftet mit „AlternateKeys“
  - Für jeden alternativen Schlüssel:
    - „Offset“: 0-basierter Byte-Offset für den alternativen Schlüssel im Datensatz.
    - „Länge“: Länge des alternativen Schlüssels in Byte.
    - „unique“: muss für den alternativen Schlüssel auf true gesetzt werden, wenn der Schlüssel keine doppelten Einträge akzeptiert, andernfalls auf false.
- wenn keine alternativen Schlüssel vorhanden sind, geben Sie eine leere Sammlung an:

```
alternateKeys: []
```

Im Folgenden finden Sie ein Beispiel für eine KSDS LISTCAT JSON-Datei.

```
{
  "catalogId": "default",
```

```
  "identifier": "AWS_M2_CARDDEMO_CARDXREF_VSAM_KSDS",
  "isIndexed": true,
  "isLinear": false,
  "isRelative": false,
  "isFixedLengthRecord": true,
  "avgRecordSize": 50,
  "maxRecordSize": 50,
  "primaryKey": {
    "offset": 0,
    "length": 16,
    "unique": true
  },
  "alternateKeys": [
    {
      "offset": 25,
      "length": 11,
      "unique": false
    }
  ]
}
```

# Konfiguration für AWS Blu Age Runtime einrichten

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 Die Geheimnisse von Blu Age Runtime
- Andere Dateien (Groovy, SQL usw.)
- Zusätzliche Webanwendung

- Eigenschaften für AWS Blu Age Runtime aktivieren
- Verfügbare Redis-Cache-Eigenschaften in AWS Blu Age Runtime
- Konfigurieren Sie die 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 Framework`Gapwalk-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-[redacted].xml
# Limits Spring logger output.
logging.level.org.springframework.beans.factory.support.DefaultListableBeanFactory : WARN
logging.level.org.springframework.statemachine : WARN
# If the datasource support mode is not static-xa, spring JTA transactions autoconfiguration must me disabled
spring.jta.enabled : false

spring:
  aws:
    client:
      datasources:
        names: primary
        primary:
          secret: arn:aws:secretsmanager:[redacted]

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

Hinweise zum Inhalt der YAML-Dateien finden Sie unter. Eigenschaften für AWS Blu Age Runtime 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 Die Geheimnisse von Blu Age Runtime

Einige der Ressourcenkonfigurationen, die Anmeldeinformationen enthalten, können mithilfe 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 Apache Tomcat im Handumdrehen abgerufen 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 die andere Art von Geheimnis verwendet werden.

YAML verweist auf Geheimnisse

`application-main.yml`Sie können auf den geheimen ARN für verschiedene Ressourcen verweisen:

JICS-Datenbank

Anmeldeinformationen für die JICS-Datenbank mit `spring.aws.jics.db.secret`

```
spring:
   aws:
     jics:
       db:
         dbname: jics
         secret: arn:aws:secretsmanager:XXXX
```

Unterstützte geheime JICS-Datenbankschlüssel:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
|---|---|
| Host | Der Hostname |
| port | Der Hafen |
| dbname | Der Name der Datenbank |
| username | Der Nutzername |
| password | Das Passwort |
| engine | Datenbank-Engine: Postgres, Oracle, Db2, Microsoft SQL Server |
| Aktuelles Schema | Spezifisches Schema, das verwendet werden soll (nur Db2-Unterstützung) |
| SSL-Verbindung | Ob die SSL-Verbindung verwendet werden soll (nur Db2-Unterstützung) |
| sslTrustStoreSpeicherort | Der Speicherort des Truststores auf dem Client (nur Db2-Unterstützung) |
| sslTrustStorePasswort | Das Passwort für den Truststore auf dem Client (nur Db2-Unterstützung) |

> **Note**
>
> Der Name der Datenbank wird entweder im Secret oder in der Yaml-Referenz angegeben. `spring.aws.jics.db.dbname`

Blusam-Datenbank

Anmeldeinformationen für die Blusam-Datenbank mit `spring.aws.client.bluesam.db.secret`

```
spring:
```

```
   aws:
     client:
       bluesam:
         db:
           dbname: bluesam
           secret: arn:aws:secretsmanager:XXXX
```

Unterstützte geheime Blusam-Datenbankschlüssel:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
|---|---|
| Host | Der Hostname |
| port | Der Hafen |
| dbname | Der Name der Datenbank |
| username | Der Nutzername |
| password | Das Passwort |
| engine | Datenbank-Engine: Postgres |

> **Note**
>
> Der Name der Datenbank wird entweder im Secret oder in der Yaml-Referenz angegeben. `spring.aws.client.bluesam.db.dbname`

Client-Datenbank

Der Client `application-profile.yml` kann auf den geheimen ARN für die Client-Datenbank verweisen. Dies erfordert eine zusätzliche Eigenschaft, um die Datenquellennamen aufzulisten. `spring.aws.client.datasources.names` `ds_name`Geben Sie für jeden Datenquellennamen den geheimen ARN in der folgenden Eigenschaft an:. `spring.aws.client.datasources.ds_name.secret` Beispiel:

```
spring:
   aws:
```

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

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.

Unterstützte geheime Schlüssel der Client-Datenbank:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
|---|---|
| Host | Der Hostname |
| port | Der Hafen |
| dbname | Der Name der Datenbank |
| username | Der Nutzername |
| password | Das Passwort |
| engine | Datenbank-Engine: Postgres, Oracle, Db2, Microsoft SQL Server |
| Aktuelles Schema | Spezifisches Schema, das verwendet werden soll (nur Db2-Unterstützung) |
| SSL-Verbindung | Ob die SSL-Verbindung verwendet werden soll (nur Db2-Unterstützung) |

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
| --- | --- |
| sslTrustStoreSpeicherort | Der Speicherort des Truststores auf dem Client (nur Db2-Unterstützung) |
| sslTrustStorePasswort | Das Passwort für den Truststore auf dem Client (nur Db2-Unterstützung) |

PGM-Dienstprogrammdatenbank

`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.

Type: Zeichenfolge

- `type`

Vollqualifizierter Name der zu verwendenden Verbindungspool-Implementierung.

Type: Zeichenfolge

Standard: `com.zaxxer.hikari.HikariDataSource`

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

Liste der Datenquellennamen.

Type: Zeichenfolge

- `dsname`
  - `dbname`

Name des Hosts.

Type: Zeichenfolge

- `secret`

Geheimer ARN der Host-Datenbank.

Type: Zeichenfolge

- `type`

Vollqualifizierter Name der zu verwendenden Verbindungspool-Implementierung.

Type: 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

- Motor (postgres/oracle/db2/mssql)
- port
- dbname
- Aktuelles Schema
- username
- password
- URL
- SSL-Verbindung
- sslTrustStoreSpeicherort
- sslTrustStorePasswort

Denn `postgres` nur der `sslMode` geheime Schlüssel value (`disable/allow/prefer/require/verify-ca/verify-full`) und die `spring.aws.rds.ssl.cert-path` YAML-Eigenschaft ermöglichen eine 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)
- sslTrustStoreSpeicherort
- sslTrustStorePasswort

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

Der geheime Wert überschreibt die YAML-Eigenschaften.

Standard Super Admin BAC und JAC

Sie können application-main.yml auch so konfigurieren, dass der Benutzername und das Passwort des standardmäßigen Super-Admin-Benutzers im Secret von AWS Secrets Manager abgerufen werden, indem Sie den ARN angeben. Das folgende Beispiel zeigt, wie dieses Geheimnis in einer YAML-Datei deklariert wird.

```
spring:
   aws:
     client:
       defaultSuperAdmin:
         secret: arn:aws:secretsmanager:XXXX
```

Unterstützte geheime Standardschlüssel für Super-Admin-Datenbanken:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
|---|---|
| username | Benutzername |
| password | Passwort |

OAuth2

Sie können `application-main.yml` auch so konfigurieren, dass das OAuth2 Client-Geheimnis abgerufen wird, indem Sie den Anbieter und den ARN angeben. AWS Secrets Manager 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 AWS Secrets Manager abgerufen. Diese Konfiguration unterstützt mehrere Anbieter, indem der Anbietername und der entsprechende geheime ARN dynamisch aufgelöst werden.

Unterstützte OAuth2 geheime Schlüssel:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
| --- | --- |
| Geheim des Kunden | Das Geheimnis, das vom Autorisierungsserv er während der Registrierung der Anwendung generiert wurde. |

Secret Manager für Redis-Caches

Die `application-main.yml` Datei kann auf den geheimen ARN für Redis-Caches verweisen. Die unterstützten sind:

- Gapwalk Redis-Anmeldeinformationen mit `spring.aws.client.gapwalk.redis.secret`
- Bluesam Redis-Anmeldeinformationen mit `spring.aws.client.bluesam.redis.secret`
- Bluesam sperrt Redis-Anmeldeinformationen mit `spring.aws.client.bluesam.locks.redis.secret`
- Datensatzkatalog Redis-Anmeldeinformationen mit `spring.aws.client.dataset.catalog.redis.secret`
- JICS Redis-Anmeldeinformationen mit `spring.aws.client.jics.redis.secret`
- Redis-Anmeldeinformationen für die Sitzung mit `spring.aws.client.jics.redis.secret`
- Redis-Anmeldeinformationen für den Sitzungs-Tracker mit `spring.aws.client.session.tracker.redis.secret`
- JICS TS Queues Redis-Anmeldeinformationen mit `spring.aws.client.jics.queues.ts.redis.secret`
- JCL Checkpoint Redis-Anmeldeinformationen mit `spring.aws.client.jcl.checkpoint.redis.secret`

- Gapwalk-Dateien sperren Redis-Anmeldeinformationen mit `spring.aws.client.gapwalk.files.locks.redis.secret`
- Blu4iv sperrt Redis-Anmeldeinformationen mit `spring.aws.client.blu4iv.locks.redis.secret`

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

```
spring:
   aws:
     client:
       gapwalk:
         redis:
           secret: arn:aws:secretsmanager:XXXX
       bluesam:
         locks:
           redis:
             secret: arn:aws:secretsmanager:XXXX
         redis:
           secret: arn:aws:secretsmanager:XXXX
       dataset:
         catalog:
           redis:
             secret: arn:aws:secretsmanager:XXXX
       jics:
         redis:
           secret: arn:aws:secretsmanager:XXXX
       session:
         tracker:
           redis:
             secret: arn:aws:secretsmanager:XXXX
       jics:
         queues:
           ts:
             redis:
               secret: arn:aws:secretsmanager:XXXX
       jcl:
         checkpoint:
           redis:
             secret: arn:aws:secretsmanager:XXXX
       gapwalk:
         files:
           locks:
```

```
            redis:
              secret: arn:aws:secretsmanager:XXXX
      blu4iv:
        locks:
          redis:
            secret: arn:aws:secretsmanager:XXXX
```

Unterstützte geheime Redis-Schlüssel:

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
| --- | --- |
| hostname | Der Hostname des Redis-Servers. |
| port | Der Redis-Serverport. |
| username | Benutzername |
| password | Passwort |

Secret Manager für SSL-Passworteinstellungen

Die `application-main.yml` Datei kann auf den geheimen ARN für SSL-Passworteinstellungen verweisen. Folgendes wird unterstützt.

- Gapwalk SSL-Anmeldeinformationen mit `spring.aws.client.ssl.secret`

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

```
spring:
   aws:
     client:
       ssl:
         secret: arn:aws:secretsmanager:XXXX
```

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
| --- | --- |
| trustStorePassword | Das Truststore-Passwort. |
| keyStorePassword | Das Keystore-Passwort. |

Secret Manager für IBM MQ-Passworteinstellungen

Die `application-main.yml` Datei kann auf den geheimen ARN für IBM MQ-Einstellungen verweisen. Folgendes wird unterstützt.

- IBM MQ-Verbindungen sind als Liste definiert, ebenso wie die Anmeldeinformationen:

  `mq.queues.jmsMQQueueManagers[N].secret:`

  N beginnt bei der ersten Verbindung bei 0.

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

```
mq.queues.jmsMQQueueManagers[0].secret: Secret-0-ARN
mq.queues.jmsMQQueueManagers[1].secret: Secret-1-ARN
```

Informationen zu Secrets ARNs Manager-Geheimnissen finden Sie unter Was ist in einem Secrets Manager Manager-Geheimnis?

Eigenschaften, die im Secret definiert sind, überschreiben ihre entsprechenden Werte in der `jmsMQ` YAML-Konfiguration.

Wenn im Secret festgelegt `queueManager` ist, überschreibt es den `mq.queues.jmsMQQueueManagers[N].jmsMQQueueManager` Wert in der YAML-Datei.

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
|---|---|
| QueueManager | Der Name des IBM MQ Queue Managers. |
| Anwendungsname | Der Name der IBM MQ-Anwendung. |
| channel | Der Name des IBM MQ-Kanals. |
| Host | Der IBM MQ-Hostname. |
| port | Der IBM MQ-Port. |
| userId | Der IBM MQ-Benutzername. |
| password | Das IBM MQ-Benutzerkennwort. |

| Geheimschlüssel | Beschreibung des geheimen Schlüssels |
| --- | --- |
| maxPoolSize | Die maximale Poolgröße von IBM MQ. |
| sslCipherKey | Die IBM MQ SSL Cipher Suite. |

## 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 Registrierung für diese Webanwendungen hängt vom Modernisierungsprojekt ab und erfolgt durch Verschieben der gewünschten WAR-Datei aus dem Ordner in den `webapps-extra` Ordner. `webapps` 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.yaml` als Konfiguration verwendet.
- `gapwalk-cl-command.war`: enthält Unterstützung für AS/400-Hilfsprogramme und wird `application-cl-command.yaml` als Konfiguration verwendet.
- `gapwalk-hierarchical-support.war`: enthält IMS/MFS-Transaktionsunterstützung und wird als Konfiguration verwendet `application-jhdb.yaml`

## Eigenschaften für AWS Blu Age Runtime 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. Auf dieser Seite erfahren Sie mehr über die verfügbaren Eigenschaften für AWS Blu Age Runtime und wie Sie sie aktivieren können.

Themen


- YML-Notation
- Schnellstart//Anwendungsfälle
- Verfügbare Eigenschaften für die Hauptanwendung
- Verfügbare Eigenschaften für optionale Webanwendungen
- Verfügbare Eigenschaften für die Client-Anwendung

## 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

# Secret manager configuration for global Redis cache
    aws:
      client:
        gapwalk:
```

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

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

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

########################################
##### Embedded Bluesam configuration #####
########################################
bluesam :
  remote : false
  cache : ehcache
  persistence : pgsql
  ehcache:
    resource-pool:
      size: 4GB
  write-behind:
    enabled: true
  pgsql :
    dataSource : bluesamDs

########################
```

```
##### Jics settings #####
#########################
rabbitmq.host: localhost
jics:
  cache: false #redis
  resource-definitions.store-type: jpa # default value: jpa, other possible value:
 redis

jics.disableSyncpoint : false
#jics.initList:
#jics.parameters.datform: DDMMYY
#jics.parameters.applid: VELOCITY
#jics.parameters.sysid: CICS
#jics.parameters.eibtrmid: TERM
#jics.parameters.userid: MYUSERID
#jics.parameters.username: MYUSERNAME
#jics.parameters.opid: XXX
#jics.parameters.cwa.length: 0
#jics.parameters.netname: MYNETNAME
#jics.parameters.jobname: MJOBNAME
#jics.parameters.sysname: SYSNAME

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

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

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

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

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

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

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

###################################
#####     SSL configuration    #####
###################################
gapwalk.ssl.enabled : true
gapwalk.ssl.trustStore : "./config/clientkey.jks"
gapwalk.ssl.trustStorePassword : mysslcertifpassword

#################################
##### MQ settings #####
#################################
mq.queues: jmsmq
mq.queues.jmsMQQueueManagers[0].jmsMQQueueManager: QM1
mq.queues.jmsMQQueueManagers[0].jmsMQAppName: Gapwalk
mq.queues.jmsMQQueueManagers[0].jmsMQChannel: DEV.APP.SVRCONN
mq.queues.jmsMQQueueManagers[0].jmsMQHost: localhost
mq.queues.jmsMQQueueManagers[0].jmsMQPort: 1415
mq.queues.jmsMQQueueManagers[0].jmsMQUserid: app
mq.queues.jmsMQQueueManagers[0].jmsMQSSLCipher: "*TLS12ORHIGHER"
mq.queues.jmsMQQueueManagers[1].jmsMQQueueManager: QM2
mq.queues.jmsMQQueueManagers[1].jmsMQAppName: Gapwalk
mq.queues.jmsMQQueueManagers[1].jmsMQChannel: DEV.APP.SVRCONN
mq.queues.jmsMQQueueManagers[1].jmsMQHost: localhost
mq.queues.jmsMQQueueManagers[1].jmsMQPort: 1415
mq.queues.jmsMQQueueManagers[1].jmsMQUserid: app

###############################
##### SQL SHIFT CODE POINT ####
###############################
```

```
# Code point 384 match unicode character \u0180
sqlCodePointShift : 384

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

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

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

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

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

#########################################################
#####  BLOCKSIZE DEFAULT VALUE  ####
#########################################################
#blockSizeDefault: 32760

#####################################
#####  JOBQUEUE CONFIGURATION    ####
#####################################
jobqueue:
  api.enabled: false
  impl: none # possible values: quartz, none
```

```
  schedulers:  # list of schedulers
    -
      name: queue1
      threadCount: 5
    -
      name: queue2
      threadCount: 5

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

########################################
#####  JCL Batch Restart Mechanism  ####
########################################
jcl:
checkpoint:
enabled: false
#expireTimeout: -1
#expireTimeoutUnit: SECONDS # Supported values: java.util.concurrent.TimeUnit
#provider: redis

----
```

- 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 | Version veröffentlichen |
|---|---|---|---|---|
| `logging.config` | Pfad | Klassenpfad: logback-main.xml | Standards chlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | ebenfalls verfügbar. | |
| `spring.jta.enabled` | boolesch | false | Standards chlüssel. Wenn der Modus zur Unterstützung der Datenquelle nicht static-xa ist, muss die automatische Konfiguration von Spring-JTA-Transaktionen deaktiviert werden. | |
| `datasource.jicsDs` + `-driver-class-name` + `-url` + `-username` + `-password` + `-type` | Standard-Spring-Datenquelle mit Unterschlüsseln | | Enthält die Verbindungsinformationen für die Jics-Datenbank. Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “JICS-Datenbank” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `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 unter erklärt. the section called “Blusam-Datenbank” | |
| `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 (). the section called “Redis-Cache-Eigenschaften” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `bluesam.maxBluesamDisablingThreadpoolSize` | Zahl | 10 | Gibt die maximale Threadpool-Größe an, die verwendet wird, um Bluesam-Datensätze für die Stapelverarbeitung zu deaktivieren. | 4.5.0 |
| `bluesam.bluesamStatusPollingInterval` | Zahl | 1000 | Gibt die Wartezeit (in Millisekunden) zwischen den einzelnen Iterationen an, wenn der Bluesam-Status abgefragt wird, um Online-Aktivitäten zu überprüfen. | 4.5.0 |
| `bluesam.maxBluesamStatusPollingRetry` | Zahl | 3 | Gibt die maximale Anzahl von Wiederholungen an, wenn die Abfrage des Bluesam-Status fehlschlägt. | 4.5.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `bluesam.checkBluesamStatus` | boolesch | false | Gibt an, ob der Status des Bluesam-Datensatzes vor dem Zugriff überprüft werden soll oder nicht. | 4.5.0 |
| `spring.aws.client.bluesam.redis.secret` | Zeichenfolge | Null | Gibt den geheimen ARN für Anmeldeinformationen für den Bluesam Redis-Cache an, siehe. the section called “AWS Die Geheimnisse von Blu Age Runtime” | |
| `spring.aws.client.bluesam.locks.redis.secret` | Zeichenfolge | Null | Gibt den geheimen ARN für Anmeldeinformationen an, damit Bluesam den Redis-Cache sperrt, siehe. the section called “AWS Die Geheimnisse von Blu Age Runtime” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `forcedDate` | Zeichenfolge | | Erzwingt das Datum auf das angegebene Datum, falls es eines gibt. | |
| `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. | |
| `blockSizeDefault` | Zahl | 32760 | Die Standardblockgröße, die für BDW-Bytes verwendet werden soll. | |
| `jics.parameters.datform` | Zeichenfolge | MMDDYY | Das Datumsformular. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.initList` | Zeichenfolge | | Die JICS-Initialisierungsliste, durch Kommas getrennt. Falls vorhanden, definiert sie kommagetrennte Namen von Listen, die beim Start von Apache Tomcat aktiviert werden sollen, in den CICS-Listen. Beispielwert:. `$UUU,DFH $IVPL,PEZ1` 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. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.parameters.applid` | Zeichenfolge | GESCHWINDIGKEIT | Die zur Identifizierung verwendete Anwendung in JICS (mindestens 4 Zeichen, keine maximale Länge). | |
| `jics.parameters.sysid` | Zeichenfolge | CICS | Die Systemidentifikation (SYSID). | |
| `jics.parameters.eibtrmid` | Zeichenfolge | BEGRIFF | Die Terminal-ID (maximal 4 Zeichen, mindestens 1 Zeichen). | |
| `jics.parameters.userid` | 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. | |
| `jics.parameters.username` | Zeichenfolge | MEIN NUTZERNAME | Der Benutzername (maximal 10 Zeichen, mindestens 1). | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.parameters.netname` | Zeichenfolge | NET/NAME | Der Netzwerkname (maximal 8 Zeichen, mindestens 1 Zeichen). | |
| `jics.parameters.opid` | Zeichenfolge | XXX | Die dreistellige Bedieneridentifikation. | |
| `jics.parameters.jobname` | Zeichenfolge | JOBNAME | Der Jobname. | |
| `jics.parameters.sysname` | Zeichenfolge | SYSNAME | Der AS4 00-Systemname (sysname). | |
| `jics.parameters.cwa.length` | Zahl | 0 | Die Länge des gemeinsamen Arbeitsbereichs (CWA). | |
| `jics.parameters.charset` | Zeichenfolge | CP037 | Weltweit verwendeter JICS-Zeichensatz. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.parameters.tsqimpl` | Zeichenfolge | Bluesam | Implementierung von JICS Temporary Storage Queue (TSQ) (zulässige Werte sind//) `bluesam` `memory` `redis` | |
| `jics.queues.ts.redis.*` | Unterstützte Redis-Eigenschaften | | Gibt die Konfigurationseigenschaften für den JICS TS Queues Redis-Server an, siehe. the section called “Unterstützte Redis-Eigenschaften” | |
| `spring.aws.client.jics.queues.ts.redis.secret` | Zeichenfolge | Null | Gibt den geheimen ARN für Anmeldeinformationen für den JICS TS Queues Redis-Server an, siehe. the section called “AWS Die Geheimnisse von Blu Age Runtime” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `lockTimeout` | Zahl | 500 | Das Sperrtimeout in Millisekunden. | |
| `sqlCodePointShift` | 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. `384` `\u0180` | |
| `sqlIntegerOverflowAllowed` | 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 | Version veröffentlichen |
|---|---|---|---|---|
| `database.cursor.overflow.allowed` | boolesch | true | Gibt an, ob der Cursorüberlauf zugelassen werden soll. Wird auf gesetzt`true`, 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. | |
| `reportOutputPath` | Zeichenfolge | `/reports` | Der Ausgabepfad für den Bericht. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `spring.session.store-type` | Zeichenfolge | Keine | Der Sitzungscache für Umgebungen mit hoher Verfügbarkeit. Mögliche Werte sind `none` oder `redis`. Die Standardeinstellung ist `none`. | |
| `stopExecutionWhenProgNotFound` | boolesch | true | 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. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `forceHR` | boolesch | false | Gibt an, ob menschenlesbares SYSPRINT entweder für die Konsolen- oder Dateiausgabe verwendet werden soll. | |
| `rollbackOnRTE` | boolesch | false | Gibt an, ob eine implizite Run-Unit-Transaktion bei Laufzeitausnahmen rückgängig gemacht werden soll. | |
| `sctThreadLimit` | long | 5 | Das Thread-Limit für das Auslösen von Skripten. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `dataSimplifier.onInvalidNumericData` | Zeichenfolge | ablehnen | Wie reagiert man, wenn man ungültige numerische Daten dekodiert? Zulässige Werte sind `reject` / `tolerate paces` / `tolerate paceslowvalues` / `tolerate ost` . Die Standardeinstellung ist `reject`. | |
| `filesDirectory` | Zeichenfolge | | Das Verzeichnis für Batch-Eingabe-/Ausgabedateien. | |
| `ims.messages.extendedSize` | boolesch | false | Gibt an, ob die erweiterte Größe für IMS-Nachrichten festgelegt werden soll. | |
| `defaultKeepExistingFiles` | boolesch | false | Gibt an, ob der vorherige Standardwert der Datenmenge festgelegt werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.db.ddlScriptLocation` | Zeichenfolge | | Der Speicherort des Jics-DDL-Skripts. Ermöglicht es Ihnen, das Jics-Datenbankschema mithilfe eines SQL-Skripts zu initiieren. Standardmäßig leer. Zum Beispiel `./jics/sql/jics.sql` . | |
| `jics.db.schemaTestQueryLocation` | Zeichenfolge | | Speicherort der SQL-Datei, die eine eindeutige Abfrage enthalten soll, die die Anzahl der Objekte im JICS-Schema (falls vorhanden) zurückgibt. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.db.dataScriptLocation` | Zeichenfolge | | Definiert den Pfad zu den SQL-Skripten, die für die Initialisierung der JICS-Datenbank verwendet werden. Akzeptiert eine durch Kommas getrennte Liste von Dateien und Verzeichnissen, sodass mehrere Skripten und Ordner angegeben werden können. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.db.dataTestQueryLocation` | 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 `jics.db.dataScriptLocation` Skript geladen, andernfalls wird das Laden der Datenbank übersprungen. | |
| `jics.data.dataJsonInitLocation` | Zeichenfolge | | | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.xa.agent.timeout` | Zahl | | | |
| `query.useConcatCondition` | boolesch | false | Gibt an, ob die Schlüsselbedingung durch Schlüsselverkettung erstellt wird oder nicht. | |
| `system.qdecfmt` | Zeichenfolge | | | |
| `disposition.checkexistence` | boolesch | false | Gibt an, ob die Überprüfung der Existenz einer Datei für Dataset mit DISP SHR oder OLD freigegeben werden soll. | |
| `useControlMVariable` | boolesch | false | Gibt an, ob die Control-M-Spezifikation für die Variablenersetzung verwendet werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `card.encoding` | Zeichenfolge | CP1145 | Kartenkodierung: zur Verwendung mit. `useControlMVariable` | |
| `mapTransfo.prefixes` | Zeichenfolge | &,@,%% | Liste der Präfixe, die bei der Transformation von ControlM-Variablen verwendet werden sollen. Jedes ist durch ein Komma getrennt. | |
| `checkinputfilesize` | boolesch | false | Gibt an, ob eine Überprüfung ausgelöst werden soll, wenn die Dateigröße ein Vielfaches der Datensatzgröße ist. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `stepFailWhenAbend` | boolesch | true | Gibt an, ob ein Ende ausgelöst werden soll, wenn die Ausführung eines Schritts fehlschlägt oder abgeschlossen wird. | |
| `bluesam.fileLoading.commitInterval` | Zahl | 100000 | Das Bluesam-Commit-Intervall. | |
| `uppercaseUserInput` | boolesch | true | Gibt an, ob Benutzereingaben in Großbuchstaben erfolgen müssen. | |
| `jhdb.lterm` | Zeichenfolge | | Ermöglicht es Ihnen, bei einer IMS-Emulation eine gemeinsame logische Terminal-ID zu erzwingen. Wenn nicht gesetzt, wird sessionId verwendet. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jhdb.identificationCardData` | Zeichenfolge | | Wird verwendet, um einige „Bediener-Identifikationsdaten“ fest in das durch den CARD-Parameter angegebene MID-Feld zu codieren. Standardmäßig leer, keine Eingabebeschränkung. | |
| `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`,,A Der Standardwert ist `CP297` | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `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`. | |
| `ims.programs` | Zeichenfolge | | Liste der zu verwenden den IMS-Programme. Trennen Sie jeden Parameter durch ein Semikolon (`;`) und jede Transaktion durch ein Komma (). `,` Beispiel: `PCP008,PCT008;PCP054,PCT054;PCP066,PCT066;PCP068,PCT068;` | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jhdb.configuration.context.encoding` | Zeichenfolge | CP297 | Die JHDB-Kodierung (Java Hierarchical Database). Erwartet eine gültige Kodierung szeichenfolgeCP1047,,,... `IBM930` `ASCII` `UTF-8` | |
| `jhdb.metadata.extrapath` | Zeichenfolge | datei:. /setup/ | Ein Konfigurationsparameter, der einen zusätzlichen, laufzeitspezifischen Stammordner für psbs- und dbds-Ordner angibt. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jhdb.checkpointPersistence` | Zeichenfolge | Keine | 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 | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | 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. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jhdb.checkpointPath` | Zeichenfolge | datei:. /setup/ | `jhdb.checkpointPersistence` Ist 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, Datenbank position und Checkpoint-Bereich) von | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | dieser Sicherung betroffen sind. | |
| `jhdb.navigation.cachenexts` | Zahl | 5000 | Die Cache-Dauer (in Millisekunden), die bei der hierarchischen Navigation für ein RDBMS verwendet wird. | |
| `jhdb.use-db-prefix` | boolesch | true | Gibt an, ob ein Datenbankpräfix in der hierarchischen Navigation für ein RDBMS aktiviert werden soll. | |
| `jhdb.query.limitJoinUsage` | boolesch | true | Gibt an, ob der Parameter Limit Join-Verwendung in RDBMS-Diagrammen verwendet werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `taskExecutor.corePoolSize` | 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. | |
| `taskExecutor.maxPoolSize` | 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. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `taskExecutor.queueCapacity` | 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 Warteschlangengröße einzurichten. (= maximale Anzahl ausstehender Transaktionen, wenn sie erreicht `taskExecutor.maxPoolSize` ist) | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `taskExecutor.allowCoreThreadTimeOut` | 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). | |
| `jics.runUnitLauncherPool.enable` | boolesch | false | Gibt an, ob der Run Unit Launcher-Pool in JICS aktiviert werden soll. | |
| `jics.runUnitLauncherPool.size` | Zahl | 20 | Die Größe des Run-Unit-Launcher-Pools in JICS. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jics.runUnitLauncherPool.validationInterval` | Zahl | 1000 | Das Intervall zwischen den einzelnen Ausführungen der Aufgabe, durch das die Größe des Pools angepasst wird. | |
| `jics.runUnitLauncherPool.parallelism` | Zahl | 2 | Die Anzahl der Threads, die verwendet werden, um die fehlenden Instanzen in der Warteschlange zu erzeugen, wenn die Anpassung saufgabe ausgeführt wird. | |
| `context.preconstruct.enable` | boolesch | false | Gibt an, ob die Vorkonstruktion des Programmkontextes aktiviert werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `context.preconstruct.frequencyInMillis` | Zahl | 100 | Das Intervall zwischen den einzelnen Ausführungen der Aufgabe, durch das die Größe des Pools angepasst wird. | |
| `context.preconstruct.parallelism` | Zahl | 5 | Die Anzahl der Threads, die verwendet werden, um die fehlenden Instanzen in der Warteschlange zu erzeugen, wenn die Anpassung saufgabe ausgeführt wird. | |
| `context.preconstruct.minInstances` | Zahl | 2 | Die Anzahl der Instanzen, die erstellt werden, wenn ein Kontext zum ersten Mal benötigt wird. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `spring.aws.application.credentials` | Zeichenfolge | Null | Laden Sie die AWS Anmeldeinformationen aus der Datei mit den Anmeldeinformationen in JICS. | |
| `jics.queues.sqs.region` | Zeichenfolge | eu-west-1 | Die AWS Region für Amazon Simple Queue Service, die in JICS verwendet wird. | |
| `jics.jcl.rt.encoding` | Zeichenfolge | CP037 | Die Kodierung der JCL-Skripte, die in die dedizierte JICS-Warteschlange geschrieben wurden. | |
| `jics.jcl.rt.queue` | Zeichenfolge | JICS | Der Name der Warteschlange, in die JCL-Skripte zur Laufzeit zeilenweise geschrieben werden können. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues.sqs.region` | Zeichenfolge | eu-west-3 | Die AWS Region für den AWS SQS MQ-Service. | |
| `quartz.scheduler.stand-by-if-error` | 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. | |
| `databaseStatistics` | boolesch | false | Gibt an, ob SQL-Builder Statistikinformationen sammeln und anzeigen dürfen. | |
| `dbDateFormat` | Zeichenfolge | yyyy-MM-dd | Das Zieldatumsformat der Datenbank. | |
| `dbTimeFormat` | Zeichenfolge | HH:MM:SS | Das DB-Zielzeitformat. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `dbTimestampFormat` | Zeichenfolge | yyyy-MM-dd hh:mm:ss.sssssss | 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`/`///EUR`EUR | |
| `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 | Version veröffentlichen |
|---|---|---|---|---|
| `localTimestampFormat` | 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-Programmen (Programmen) verwendete Zeitformat. | |
| `pgmTimestampFormat` | Zeichenfolge | yyyy-MM-dd-hh.mm.ss.ssssss | Das Zeitstempelformat. | |
| `cacheMetadata` | boolesch | true | Gibt an, ob Datenbank-Metadaten zwischengespeichert werden sollen. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `forceDisableSQLTrimStringType` | boolesch | false | Gibt an, ob das Kürzen aller SQL-Zeichenkettenparameter deaktiviert werden soll. | |
| `fetchSize` | Zahl | | Der FetchSize-Wert für Cursor. Wird verwendet, wenn Daten mithilfe von Chunks mithilfe von Load/Unload-Utils abgerufen werden. | |
| `check-groovy-file` | boolesch | true | Gibt an, ob der Inhalt von Groovy-Dateien vor der Registrierung überprüft werden soll. | |
| `qtemp.uuid.length` | Zahl | 9 | Die Länge der eindeutigen QTEMP-ID. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `qtemp.dblog` | boolesch | false | Ob die QTEMP-Datenbankprotokollierung aktiviert werden soll. | |
| `qtemp.cleanup.threshold.hours` | Zahl | 0 | Um anzugeben, wann aktiviert `qtemp.dblog` ist. Die Lebensdauer der DB-Partition (in Stunden). | |
| `sort.function` | Zeichenfolge | | Der Name der Sortierfunktion für die blu4iv-Datenbank. | |
| `invalidDataToleranceDaten` | boolesch | true | Gibt an, ob ungültige für den gepackten Typ toleriert werden. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `program.timeout` | 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. | |
| `gapwalk.line.separator` | 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 line.separatator wird verwendet. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `enableActivePgmIdCache` | 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 | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues.default.syncpoint` | boolesch | false | Definiert das Standardverhalten für MQ PUT-Befehle, wenn weder MQPMO_SYNCPOINT noch MQPMO_NO_SYNCPOINT gesetzt sind. Wenn es auf true gesetzt ist, verhält es sich so, als ob Nachrichten während des PUT-Befehls NICHT direkt festgeschrieben werden. `MQPMO_SYNCPOINT` Wenn der Wert auf False gesetzt ist, verhält es sich so, als `MQPMO_NO_SYNCPOINT` ob Nachrichten während des PUT-Befehls direkt festgeschrieben werden. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `dataSimplifier.byteRangeBoundsCheck` | boolesch | false | Wenn der Wert auf true gesetzt ist, ByteRange wird sichergestellt, dass keine Daten mit falschen Werten erstellt werden. Der Standardwert lautet „false“. | |
| `file.stdoutIntoLogger` | 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. | |
| `tempFilesDirectory` | Zeichenfolge | Null | Gibt den Namen des Speicherorts der temporären Dateien an, die generiert werden. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
| --- | --- | --- | --- | --- |
| `cleanTempFilesDirectoryAtStartup` | boolesch | true | Gibt an, ob der Inhalt des Ordners für temporäre Dateien beim Start der Anwendung gelöscht werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `tempFolderPattern` | Zeichenfolge | Null | 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.<br><br>HOST: der Hostname.<br><br>JOBID: Die ID des Jobs.<br><br>HASHCODE: Der Hashcode des Job-Kontextes.<br><br>TIMESTAMP: das Muster, das beim Abrufen des Zeitstempels verwendet werden soll. Der Zielname des temporären Ordners ist TMP_DIR_ | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | {}. tempFolderPattern Im Fall des folgenden Musters beginnt der Name beispielsweise mit der Job-ID und endet mit dem „Zeitstempel“ tempFolderPattern: 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 () (). DefaultJobContext | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `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. | |
| `jics.spool.smtp.username` | Zeichenfolge | Null | Gibt den Benutzernamen des SMTP-Servers an. | |
| `jics.spool.smtp.debug` | boolesch | false | Gibt den Debug-Modus für den SMTP-Server an. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk-application.security` | Zeichenfolge | `disabled` | Schaltet die globale Sicherheitskonfiguration um (XSS, CORS, CSRF, OAUTH-Authentifizierung...). Zulässige Werte sind `disabled` und `enabled`. | |
| `gapwalk-application.identity` | Zeichenfolge | Null | Globale Authentifizierungsmethode. Der empfohlene Wert ist `oauth`. Zulässige Werte sind `json` und `oauth`. Diese Option ist erforderlich, wenn sie `gapwalk-application.security` ist `enabled`. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk-application.security.issuerUri` | Zeichenfolge | Null | Die Aussteller-URI des Identitätsanbieters (IdP). Diese Option ist erforderlich, wenn sie `gapwalk-application.identity` ist. `oauth` | |
| `gapwalk-application.security.allowedOrigins` | Zeichenfolge [] | `null` | Die Liste der Ursprünge, die zugelassen werden sollen. Diese Option `gapwalk-application.identity` muss auf eingestellt seinoauth. | |
| `gdgDirectoryPath` | Zeichenfolge | `output/gdg` | Der GDG-Verzeichnispfad ist das Verzeichnis, in dem die GDG-Dateien gespeichert sind. | 4.6.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk-application.security.claimGroupName` | Zeichenfolge | `cognito:groups` | Das Claim-Attribut, das die Liste aller Gruppen enthält, zu denen ein Benutzer gehört. Verwenden Sie `cognito:groups` für Amazon Cognito oder eine andere Zeichenfolge für einen ausländischen IdP. | |
| `gapwalk-application.security.userAttributeName` | Zeichenfolge | `username` | Der Name des Anspruchsattributs, der zur Identifizierung einer Benutzeranfrage verwendet wird. Verwenden Sie `username` für Amazon Cognito, `preferred_username` für Keycloak oder eine andere Zeichenfolge für einen ausländischen IdP. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk-application.security.localhostWhitelistingEnabled` | boolesch | true | Gibt an, ob die Authentifizierung anhand beliebiger Anfragen aktiviert werden soll. `localhost` | |
| `gapwalk-application.defaultSuperAdminUserName` | Zeichenfolge | `sadmin` | Wenn deaktiviert `gapwalk-application.security` ist, gibt den lokalen Standard-Superbenutzernamen an. | |
| `gapwalk-application.defaultSuperAdminUserPwd` | Zeichenfolge | `sadmin` | Wenn deaktiviert `gapwalk-application.security` ist, gibt das standardmäßige lokale Superuser-Passwort an. | |
| `gapwalk-application.security.filterURIs` | Zeichenfolge | `disabled` | Schaltet die URIs Filterkonfiguration um. Zulässige Werte sind `disabled` und `enabled`. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk-application.security.blockedURIs` | Zeichenfolge [] | Null | Die Liste der URIs zu blockierenden. Diese Option ist erforderlich, wenn sie `gapwalk-application.security.filterURIs` ist`enabled`. | |
| `jics.redis.*` | Unterstützte Redis-Eigenschaften | | Gibt die Konfigurationseigenschaften für die JICS Redis Server Connection Factory an, siehe. the section called “Unterstützte Redis-Eigenschaften” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `spring.aws.client.jics.redis.secret` | Zeichenfolge | Null | Gibt den geheimen ARN für Anmeldeinformationen für die JICS Redis Server Connection Factory an, siehe. the section called “AWS Die Geheimnisse von Blu Age Runtime” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jcl.checkpoint.enabled` | boolesch | false | Gibt an, ob der JCL-Checkpoint-Mechanismus aktiviert ist, um einen Neustart des Jobs zu ermöglichen. JCL-Checkpoints werden zu Beginn jedes Schritts oder Hauptprogrammaufrufs erstellt und in der speicherinternen Registrierung gespeichert. Alle Checkpoints auf Step-Ebene werden am Ende des Jobs dauerhaft gespeichert, sofern ein Persistenzanbieter definiert ist. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jcl.checkpoint.expireTimeout` | Zahl | -1 | Gibt die Zeitdauer für die Beibehaltung von JCL-Checkpoints im Persistenzanbieter oder in der In-Memory-Registrierung an. | |
| `jcl.checkpoint.expireTimeoutUnit` | Zeichenfolge | SEKUNDEN | Gibt die Einheit für die Dauer der `jcl.checkpoint.expireTimeout` Eigenschaft an. Unterstützte Enum-Konstantenwerte: java.util.concurrent. TimeUnit. | |
| `jcl.checkpoint.provider` | Zeichenfolge | Null | Gibt den Persistenzanbieter für den JCL-Checkpoint-Mechanismus an. Zulässige Werte sind. `redis` | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `jcl.checkpoint.redis.*` | Unterstützte Redis-Eigenschaften | | Gibt die Konfigurationseigenschaften für den REDIS-Persistenzanbieter des JCL-Checkpoint-Mechanismus an, siehe. the section called “Unterstützte Redis-Eigenschaften” | |
| `spring.aws.client.jcl.checkpoint.redis.secret` | Zeichenfolge | Null | Gibt den geheimen ARN für Anmeldeinformationen für den Redis-Persistenzanbieter des JCL-Checkpoint-Mechanismus an, siehe. the section called “AWS Die Geheimnisse von Blu Age Runtime” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk.ssl.enabled` | boolesch | false | Wird angezeigt, um die folgenden `gapwalk.ssl.*` Eigenschaften auf die aktuellen JVM-Systemeigenschaften zu setzen, sofern sie nicht bereits beim Start der Anwendung festgelegt wurden. | |
| `gapwalk.ssl.trustStore` | Zeichenfolge | Null | Setzen Sie den Wert auf die Systemeigenschaft, `javax.net.ssl.trustStore` falls dieser Wert nicht bereits beim Start der Anwendung festgelegt wurde. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk.ssl.trustStorePassword` | Zeichenfolge | Null | Setzen Sie den Wert auf die Systemeigenschaft, `javax.net.ssl.trustStorePassword` falls diese nicht bereits beim Start der Anwendung eingerichtet wurde. Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “Secret Manager für SSL-Passworteinstellungen” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk.ssl.trustStoreType` | Zeichenfolge | Null | Setzen Sie den Wert auf die Systemeigenschaft, `javax.net.ssl.trustStoreType` falls diese nicht bereits beim Start der Anwendung eingerichtet wurde. | |
| `gapwalk.ssl.keyStore` | Zeichenfolge | Null | Setzen Sie den Wert auf die Systemeigenschaft, `javax.net.ssl.keyStore` falls diese nicht bereits beim Start der Anwendung eingerichtet wurde. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `gapwalk.ssl.keyStorePassword` | Zeichenfolge | Null | Setzen Sie den Wert auf die Systemeigenschaft, `javax.net.ssl.keyStorePassword` falls diese nicht bereits beim Start der Anwendung eingerichtet wurde. Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “Secret Manager für SSL-Passworteinstellungen” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues` | Zeichenfolge | `sqs` | Gibt an, welcher unterstützte Queue-Broker verwendet werden soll, wenn Amazon SQS `sqs` verwendet wird, Rabbit MQ vor Ort `rabbitmq` verwendet wird oder ob IBM MQ vor Ort verwendet wird. `jms` | |
| `mq.queues.jmsMQQueueManagers[N]` | | | Wenn `mq.queues` ist`jms`, ermöglicht die Angabe einer IBM MQ-Verbindungsliste. `mq.queues.jmsMQQueueManagers[0]` für die erste Verbindung, `mq.queues.jmsMQQueueManagers[1]` für die zweite usw. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues.jmsMQQueueManagers[N].jmsMQQueueManager` | Zeichenfolge | Null | Der Name des IBM MQ-Wartes chlangenmanagers. | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQAppName` | Zeichenfolge | Null | Der Name der IBM MQ-Anwendung. | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQChannel` | Zeichenfolge | Null | Der Name des IBMMQ-Kanals. | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQHost` | Zeichenfolge | Null | Der IBM MQ-Hostname. | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQPort` | Zahl | Null | Der IBMMQ-Port. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues.jmsMQQueueManagers[N].jmsMQUserid` | Zeichenfolge | Null | Der IBMMQ-Benutzername. | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQPassword` | Zeichenfolge | Null | Das IBMMQ-Benutzerkennwort. Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “Secret Manager für IBM MQ-Passworteinstellungen” | |
| `mq.queues.jmsMQQueueManagers[N].jmsMQMaxPoolSize` | Zahl | 0 | Die maximale Poolgröße von IBM MQ. Bei 0 sind unendlich viele physische Verbindungen aktiviert. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mq.queues.jmsMQQueueManagers[N].jmsMQSSLCipher` | Zeichenfolge | Null | Die IBM MQ SSL Cipher Suite. Ein Beispiel könnte sein. `"*TLS120R HIGHER"` Weitere Informationen finden Sie in der offiziellen Dokumentation TLS CipherSpecs und CipherSuites in IBM MQ-Klassen für JMS. | |
| `mq.queues.non.jms.client` | boolesch | false | Geben Sie an, ob es sich bei dem Zielclient, an den Nachrichten gesendet werden sollen, nicht um JMS handelt. Das native MQ-Format wird für Nicht-JMS-Clients verwendet, während RFH2 das Format für JMS verwendet wird. | 4.5.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | Wann `mq.queues` ist der IBM `rabbitmq` MQ-Hostname. | |
| `mq.queues.rabbitMQHost` | | | Der Rabbit MQ-Hostname. | |
| `mq.queues.rabbitMQVirtualHost` | | | Der virtuelle Hostname von Rabbit MQ. | |
| `mq.queues.rabbitMQPort` | | | Der Rabbit MQ-Port. | |
| `mq.queues.rabbitMQUsername` | | | Der Rabbit MQ-Benutzer. | |
| `mq.queues.rabbitMQPassword` | | | Das Rabbit MQ-Passwort. | |
| `mf.runtime.switch.N` | boolesch | true | Aktiviert das Einfügen von Nullen für zeilenseq uenzielle MF-Nature-Dateien. | 4.4.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `mf.runtime.switch.T` | boolesch | false | Ermöglicht das Einfügen von Tabulatorzeichen in zeilenseq uenzielle MF-Nature-Dateien. | 4.4.0 |
| `gapwalk.database.support.useSavePointToRestoreFail` | boolesch | false | Ermöglicht die Wiederherstellung von Transaktionen im Falle eines Fehlers mithilfe von Savepoints bei Einfügeabfragen. Die Aktivierung dieser Eigenschaft kann sich auf die Datenbankleistung auswirken. Sie können diese Einstellung für bestimmte Abfragen mithilfe der query-to-database Mapping-Konfiguration überschreiben. | 4.6.0 |

## 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. The following tables contain lists of the available key/value Parameter für die Konfiguration jeder optionalen Webanwendung unterstützen.

gapwalk-utility-pgm.war

Diese optionale Webanwendung enthält Unterstützung für Z/OS-Hilfsprogramme.

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

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `logging.config` | Pfad | Klassenpfad: logback-utility.xml | Standards chlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar. | |
| `spring.jta.enabled` | boolesch | false | Standards chlüssel. Wenn der Datenquellenunterstützungsmodus nicht static-xa ist, muss die auto Konfiguration | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | 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. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `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.<br><br>Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “Client-Datenbank” | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `encoding` | Zeichenfolge | ASCII | Die in Hilfsprogrammen verwendete Kodierung. Erwartet eine gültige Kodierung `CP1047IBM930,,A` | |
| `sysPunchEncoding` | Zeichenfolge | ASCII | Der Zeichensatz für die Syspunch-Kodierung. Erwartet eine gültige Kodierung `CP1047,,IBM930,A` `UTF-8` | |
| `systin.encoding` | Zeichenfolge | ASCII | Der Kodierungszeichensatz des SYSTIN−Dateidatensatzes. Erwartet eine gültige Kodierung `CP1047,IBM930,A` | 4.5.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `zonedMode` | Zeichenfolge | EBCDIC_STRICT | Der Modus für die Kodierung oder Dekodierung von Datentypen in Zonen. Zulässige Werte sind`EBCDIC_STRICT //EBCDIC_MODIFIED . AS400` | |
| `idcams.encoding.forced` | Zeichenfolge | | Die im IDCAMS-Hilfsprogramm verwendete Kodierung. Erwartet eine gültige Kodierung `CP1047,IBM930,A` | 4.4.0 |
| `unload.chunkSize` | Zahl | 0 | Die für das Hilfsprogramm zum Entladen verwendete Blockgröße. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `unload.computeRecordSizeIfNull` | boolesch | false | Legt fest, ob die Datensatzgröße berechnet werden soll, wenn sie nicht angegeben ist. Falls angegeben, bleibt der Wert unverändert. | |
| `unload.sqlCodePointShift` | Zahl | 0 | Das SQL-Code-Hilfsprogramm Pointshift for Unload. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank Postgresql DB2 ist. | |
| `unload.columnFiller` | Zeichenfolge | Leerzeichen | Der Spaltenfüller des Hilfsprogramms zum Entladen. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `unload.varCharIsNull` | boolesch | false | Verwenden Sie diesen Parameter im INFTILB-Programm. Wenn er auf gesetzt ist, `true` wird für alle Felder, die keine NULL-Werte zulassen, mit leeren Werten (Leerzeichen) eine leere Zeichenfolge zurückgegeben. | |
| `unload.useDatabaseConfiguration` | boolesch | false | Gibt an, ob die Datums- oder Uhrzeitkonfiguration aus application-main.yml im Unload-Hilfsprogramm verwendet werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `unload.format.date` | Zeichenfolge | MM/dd/yyyy | Falls aktiviert, `unload.useDatabaseConfiguration` das Datumsformat, das im Entladeprogramm verwendet werden soll. | |
| `unload.format.time` | Zeichenfolge | Hh.mm.ss | Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitformat, das im Entladeprogramm verwendet werden soll. | |
| `unload.format.timestamp` | Zeichenfolge | yyyy-MM-dd-HH.mm.ss.ssssss | Falls aktiviert, `unload.useDatabaseConfiguration` das Zeitstempelformat, das im Entladeprogramm verwendet werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `unload.nbi.whenNull` | hexadezimal | 6F | Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank Null ist. | |
| `unload.nbi.whenNotNull` | hexadezimal | 00 | Der Wert des Null-Byte-Indikators (NBI), der hinzugefügt werden soll, wenn der Wert aus der Datenbank nicht Null ist. | |
| `unload.nbi.writeNullIndicator` | boolesch | false | Gibt an, ob der Null-Indikator in die Entladeausgabedatei geschrieben werden soll. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `unload.bmc.useInto` | boolesch | false | Gibt an, ob das INTO-BMC-Steuerschlüsselwort für das Unload-Hilfsprogramm verarbeitet werden soll. | |
| `unload.fetchSize` | Zahl | 0 | Ermöglicht es Ihnen, die Abrufgröße bei der Verarbeitung von Cursorn im Hilfsprogramm zum Entladen einzustellen. | |
| `unload.noPad` | boolesch | true | Gibt an, dass Felder mit variabler Länge (VARCHAR) ohne Auffüllung auf die maximale Länge entladen werden sollen. | 4.5.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `treatLargeNumberAsInteger` | boolesch | false | Gibt an, ob große Zahlen als behandelt werden sollen. `Integer` Sie werden `BigDecimal` standardmäßig wie behandelt. | |
| `load.batchSize` | Zahl | 0 | Die Batchgröße des Ladediensprogramms. | |
| `load.format.localDate` | Zeichenfolge | dd.mm.yyyy\\ yyyy-mm-dd dd/MM/yyyy | Das zu verwendende lokale Datumsformat des Load Utilitys. | |
| `load.format.localTime` | Zeichenfolge | hh:mm:ss\ hh.mm.ss | Das zu verwendende lokale Zeitformat des Load Utilitys. | |
| `load.format.dbDate` | Zeichenfolge | yyyy-MM-dd | Das zu verwendende Load Utility-Datenbankformat. | |
| `load.format.dbTime` | Zeichenfolge | hh:mm:ss | Die zu verwendende Uhrzeit der Load Utility-Datenbank. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `load.sqlCodePointShift` | Zahl | 0s | Das SQL-Code-Hilfsprogramm Pointshift für Load. Führt den Prozess zum Verschieben von Zeichen aus. Erforderlich, wenn Ihre Zieldatenbank Postgresql DB2 ist. | |
| `load.applyRollback` | boolesch | false | Legen Sie diesen Parameter fest, `true` um anzugeben, dass der Dienst Tabellenänderungen rückgängig machen soll, wenn beim Laden von Daten in die Datenbank ein Fehler auftritt. | |
| `forcedDate` | Zeichenfolge | | Erzwingt das Datum auf das angegebene Datum, falls es eines gibt. | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `frozenDate` | boolesch | true | Gibt an, ob das Datum eingefroren werden soll. Gilt nur, wenn `forcedDate` es auch gesetzt ist. | |
| `jcl.type` | Zeichenfolge | mvs | .jcl-Dateityp. 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. | |
| `hasGraphic` | boolesch | false | Ob das INFUTILB-Hilfsprogramm GRAPHIC-Spalten verarbeiten muss. DB2 | |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `convertGraphicDataToFullWidth` | boolesch | true | Gibt an, ob Grafikdaten in das Format mit voller Breite konvertiert werden sollen. | |

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 |
|---|---|---|---|
| `logging.config` | Pfad | Klassenpfad:logback-utility.xml | Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar. |
| `spring.jta.enabled` | boolesch | false | Standardschlüssel. Wenn der Datenquellenunterstützungsmodus nicht static-xa ist, muss die auto Konfiguration von Spring JTA-Transaktionen deaktiviert werden. |

| Schlüssel | Typ | Standardwert | Beschreibung |
| --- | --- | --- | --- |
| `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.<br><br>Alternativ wird die Verwendung von AWS Geheimnissen dringend empfohlen, wie unter erklärt. the section called “Client-Datenbank” |
| `encoding` | Zeichenfolge | ASCII | Die in Hilfsprogrammen verwendete Kodierung. Erwartet eine gültige Kodierung `CP1047IBM930,,ASCII,UTF-8` |

| Schlüssel | Typ | Standardwert | Beschreibung |
|---|---|---|---|
| zonedMode | Zeichenfolge | EBCDIC_STRICT | Der Modus für die Kodierung oder Dekodierung von Datentypen in Zonen. Zulässige Werte sindEBCDIC_ST RICT //EBCDIC_MO DIFIED .AS400 |
| 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 | | Verbindet das Datum mit dem angegebenen Datum, 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 |
| --- | --- | --- | --- |
| `logging.config` | Pfad | Klassenpfad:logback-utility.xml | Standardschlüssel für den Verweis auf die Logback-Konfigurationsdatei. Andere Standard-Logging-Schlüssel sind ebenfalls verfügbar. |
| `spring.jta.enabled` | boolesch | false | Standardschlüssel. Wenn der Datenquellenunterstützungsmodus nicht static-xa ist, muss die auto Konfiguration von Spring JTA-Transaktionen deaktiviert werden. |
| `jhdb.configuration.context.encoding` | Zeichenfolge | | Die JHDB-Kodierung (Java Hierarchical Database). Erwartet eine gültige Kodierungszeichenfolge`CP1047`,,,... `IBM930` `ASCII` `UTF-8` |

| Schlüssel | Typ | Standardwert | Beschreibung |
| --- | --- | --- | --- |
| `jhdb.checkpointPersistence` | Zeichenfolge | Keine | Der Checkpoint-Persistenzmodus. Zulässige Werte sindnone//add. end Wird verwendetadd, 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. |

## Verfügbare Eigenschaften für die Client-Anwendung

Für Ihre modernisierte Anwendung sind möglicherweise spezielle Eigenschaftskonfigurationen für die Spring-Clientanwendung erforderlich. Diese Eigenschaften initialisieren Beans aus Klassen, die in Runtime-JAR-Dateien verpackt sind. In der `application-profile.yaml` Datei, in der der Profilwert während der Anwendungsgenerierung festgelegt wird, können Sie diese Eigenschaften konfigurieren. In der folgenden Tabelle sind die Schlüssel-/Wertparameter aufgeführt, die für die Konfiguration der Client-Webanwendung verfügbar sind, die Beans aus Klassen verwendet, die in der Gapwalk-Runtime verpackt sind

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| `blu4iv.dtaara.library.disable` | boolesch | false | Steuert die Nutzung der Bibliothek im Kontext von Datenbereichsoperationen. Wenn dieser Wert auf true gesetzt ist, ist die Bibliotheksnutzung für Datenbereichsoperationen deaktiviert, dies hat jedoch keinen Einfluss auf die Verwendung von QTemp. Wenn der Wert auf false gesetzt ist, wird die Bibliothek bei der Ausführung | 4.5.0 |

| Schlüssel | Typ | Standardwert | Beschreibung | Version veröffentlichen |
|---|---|---|---|---|
| | | | von CRUD-Operationen für den Datenbereich berücksichtigt. | |

## Verfügbare Redis-Cache-Eigenschaften in AWS Blu Age Runtime

In diesem Dokument erfahren Sie mehr über die Redis-Caches in AWS Blu Age Runtime sowie über die Gapwalk-Konfiguration, die unterstützten Redis-Eigenschaften und darüber, wie `application-main.yml` Dateien auf geheime ARN für Redis-Caches verweisen können.

### Redis-Caches in Blu Age Runtime AWS

Redis-Server können als Caches für verschiedene Funktionen in der AWS Blu Age Gapwalk-Anwendung verwendet werden, z. B.:

| AWS Blu Age Runtime-Funktionen, die Redis-Caching verwenden | Beschreibung |
|---|---|
| Bluesam-Cache | Ein Redis Blusam-Cache zum effizienten Lesen von Datensätzen mithilfe einer Write-Behind-Strategie, um schreibintensive Workloads zu optimieren, die bei Batch-Payloads auftreten. |
| Blusam-Sperren | Ein Cache für verteilte Sperren für Datensätze und Datensätze. |
| Datensatz-Katalog | Der Catalog-Dataset-Cache. |
| Sitzungs-Cache | Ein Redis-Cache für HttpSession. Der Cache speichert den Benutzernamen, den Status des Dialogs mit dem Angular-Frontend und spezifische Dialektinformationen (BMS, MFS, 00). AS4 |

| AWS Blu Age Runtime-Funktionen, die Redis-Caching verwenden | Beschreibung |
| --- | --- |
| Sitzungs-Tracker | Ein Cache mit aktiven Sitzungen mit zugehörigem Benutzernamen und session-creation-time Informationen. |
| JICS-Cache | Ein Cache für JICS-Ressourcendefinitionen. |
| TS-Warteschlangen | Speicher für TS-Warteschlangen. |
| JCL-Checkpoint | JCL-Checkpoint-Cache. |
| Gapwalk-Dateisperren | Ein Cache für verteilte Dateisperren nach Job. |
| Blu4IV-Sperren | Speicher für Blu4IV-Datensatzsperren. |

## Redis Gapwalk-Konfiguration

Die globale Redis-Konfiguration wird verwendet, wenn sie als Caching-Mechanismus angegeben `redis` ist und keine Redis-Konfiguration für die spezifische Funktion bereitgestellt wird. Diese Konfiguration ermöglicht es Ihnen, dieselbe Konfiguration für mehrere Redis-Caches gleichzeitig zu verwenden.

Im folgenden Beispiel verwenden der Blusam-Dataset-Cache und der JICS-Cache die Konfiguration `gapwalk.redis` (`redis.server1`), da ihr Cachetyp auf gesetzt ist und unter und keine impliziten `redis` Redis-Eigenschaften angegeben sind. the section called “Definitionen von JICS-Ressourcen” the section called “Definitionen von JICS-Ressourcen” Der Blusam-Locks-Cache verwendet jedoch eine andere Redis-Konfiguration ()`redis.server2`, da seine Redis-Eigenschaften explizit definiert sind.

```
...

 gapwalk:
   redis:
     hostName: redis.server1
  port: 6379
 ...

 bluesam:
```

```
  # Redis bluesam cache
  cache: redis
  # Redis locks cache
  locks:
    cache: redis
 hostName: redis.server2
 port: 6379
...
# Redis jics cache
jics:
  resource-definitions:
    store-type: redis
...
```

Um die globale Redis-Konfiguration zu aktivieren, fügen Sie die folgende Konfiguration unter hinzu. `main-application.yml`

```
gapwalk:
  redis:
    hostName: localhost
    port: 6379
    mode: standalone                        # Optional
    username:                               # Optional
    password: ""                            # Optional
    useSsl: false                           # Optional
    database: 0                             # Optional
    maxTotal: 128                           # Optional
    maxIdle: 128                            # Optional
    minIdle: 16                             # Optional
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

## Unterstützte Redis-Eigenschaften

Die folgende Tabelle zeigt die Redis-Eigenschaften, die für globale und spezifische Redis-Caches auf AWS Blu Age Runtime unterstützt werden.

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `mode` | Nein | Der Redis-Laufmodus. | `standalone \| cluster` | `standalone` |
| `hostname` | Ja | Der Hostname oder die IP-Adresse des Redis-Servers. | Zeichenfolge | Null |
| `port` | Ja | Die Portnummer, auf der der Redis-Server auf Verbindungen wartet. | int | Null |
| `username` | Nein | Der Benutzername für die Authentifizierung. | Zeichenfolge | Null |
| `password` | Nein | Das Passwort für die Authentifizierung. | Zeichenfolge | leere Zeichenfolge |
| `useSsl` | Nein | Gibt an, ob die SSL/TLS-Verschlüsselung für die Redis-Ver | boolesch | false |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| | | bindung aktiviert werden soll. | | |
| `database` | Nein | Die zu verwenden de Redis-Dat enbanknummer. Redis unterstützt mehrere logische Datenbank en, und diese Eigenschaft gibt an, welche verwendet werden soll. | int | 0 |
| `maxTotal` | Nein | Die maximale Anzahl von Verbindungen, die im Redis-Ver bindungspool zulässig sind. | int | 128 |
| `maxIdle` | Nein | Die maximale Anzahl inaktiver Verbindungen, die im Redis-Ver bindungspool zulässig sind. | int | 128 |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `minIdle` | Nein | Die Mindestanzahl inaktiver Verbindungen, die im Redis-Verbindungspool aufrechterhalten werden sollen. | int | 16 |
| `testOnBorrow` | Nein | Ein boolescher Wert, der angibt, ob Verbindungen validiert werden sollen, bevor sie aus dem Pool ausgeliehen werden. | boolesch | true |
| `testOnReturn` | Nein | Ein boolescher Wert, der angibt, ob Verbindungen validiert werden sollen, bevor sie an den Pool zurückgegeben werden. | boolesch | true |
| `testWhileIdle` | Nein | Ein boolescher Wert, der angibt, ob inaktive Verbindungen im Pool regelmäßig überprüft werden sollen. | boolesch | true |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `testOnCreate` | Nein | Ein boolescher Wert, der angibt, ob Verbindungen bei ihrer Erstellung validiert werden sollen. | boolesch | true |
| `minEvictableIdleTimeMillis` | Nein | Die Mindestdauer (in Millisekunden), die eine inaktive Verbindung im Pool verbleiben muss, bevor sie entfernt werden kann. | long | 60000 L |
| `timeBetweenEvictionRunsMillis` | Nein | Die Zeit (in Millisekunden) zwischen aufeinanderfolgenden Läufen des Verbindungs-Evictor-Threads im Leerlauf. | long | 30000L |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `numTestsPerEvictionRun` | Nein | Die maximale Anzahl von Verbindungen, die bei jedem Lauf des Verbindungs-Evictor-Threads im Leerlauf getestet werden sollen. | int | -1 |
| `blockWhenExhausted` | Nein | Ein boolescher Wert, der angibt, ob eine Verbindung blockiert und darauf gewartet werden soll, dass eine Verbindung verfügbar wird, wenn der Pool erschöpft ist. | boolesch | true |
| `nettyThreads` | Nein | Die Anzahl der Netty-Threads, die für die Verarbeitung von Redis-Verbindungen verwendet werden sollen. | int | 32 |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `subscriptionsPerConnection` | Nein | Die maximale Anzahl von Abonnements, die pro Redis-Verbindung zulässig sind. | int | 10 |
| `subscriptionConnectionPoolSize` | Nein | Die maximale Anzahl von Verbindungen, die im Redis-Abonnement-Verbindungspool zulässig sind. | int | 100 |
| `pageSizeInBytes` | Nein | Die Standard-Seitengröße in Byte für Redis-Operationen. | long | 262144000 |
| `readTimeout` | Nein | Das Lese-Timeout in Millisekunden für Redis-Operationen. | long | 2000 |

| Name der Eigenschaft | Erforderlich? | Beschreibung | Werte | Standard |
|---|---|---|---|---|
| `timeToLiveMillis` | Nein | Die Dauer (in Millisekunden), für die ein Cache-Eintrag im Cache verbleibt, bevor er als abgelaufen betrachtet und entfernt wird. Wenn diese Eigenschaft nicht angegeben ist, laufen Cache-Einträge standardmäßig nicht automatisch ab. | long | -1 |

## Redis-Cache-Eigenschaften

### Redis Blusam-Cache

```
bluesam:
   cache: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                          # Optional
     username:                                 # Optional
     password: ""                              # Optional
     useSsl: false                             # Optional
     database: 0                               # Optional
     maxTotal: 128                             # Optional
     maxIdle: 128                              # Optional
     minIdle: 16                               # Optional
```

```
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
    timeToLiveMillis: 60000                 # Optional
```

### Redis Blusam-Cache

```
bluesam:
   locks:
     cache: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
       hostName: localhost
       port: 6379
       mode: standalone                       # Optional
       username:                              # Optional
       password: ""                           # Optional
       useSsl: false                          # Optional
       database: 0                            # Optional
       maxTotal: 128                          # Optional
       maxIdle: 128                           # Optional
       minIdle: 16                            # Optional
       testOnBorrow: true                     # Optional
       testOnReturn: true                     # Optional
       testWhileIdle: true                    # Optional
       testOnCreate: true                     # Optional
       minEvictableIdleTimeMillis: 60000      # Optional
       timeBetweenEvictionRunsMillis: 30000   # Optional
       numTestsPerEvictionRun: -1             # Optional
       blockWhenExhausted: true               # Optional
       nettyThreads: 32                       # Optional
       subscriptionsPerConnection: 10         # Optional
       subscriptionConnectionPoolSize: 100    # Optional
```

```
      pageSizeInBytes: 8192                  # Optional
      readTimeout: 2000                      # Optional
```

### Sitzungs-Cache

```
spring:
  session:
    store-type: redis
# If the following redis properties are not specified gapwalk.redis configuration will
be used for this cache
jics:
  redis:
    hostName: localhost
    port: 6379
    mode: standalone                        # Optional
    username:                               # Optional
    password: ""                            # Optional
    useSsl: false                           # Optional
    database: 0                             # Optional
    maxTotal: 128                           # Optional
    maxIdle: 128                            # Optional
    minIdle: 16                             # Optional
    testOnBorrow: true                      # Optional
    testOnReturn: true                      # Optional
    testWhileIdle: true                     # Optional
    testOnCreate: true                      # Optional
    minEvictableIdleTimeMillis: 60000       # Optional
    timeBetweenEvictionRunsMillis: 30000    # Optional
    numTestsPerEvictionRun: -1              # Optional
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

### Definitionen von JICS-Ressourcen

```
jics:
   resource-definitions:
     store-type: redis
```

```
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                        # Optional
     username:                               # Optional
     password: ""                            # Optional
     useSsl: false                           # Optional
     database: 0                             # Optional
     maxTotal: 128                           # Optional
     maxIdle: 128                            # Optional
     minIdle: 16                             # Optional
     testOnBorrow: true                      # Optional
     testOnReturn: true                      # Optional
     testWhileIdle: true                     # Optional
     testOnCreate: true                      # Optional
     minEvictableIdleTimeMillis: 60000       # Optional
     timeBetweenEvictionRunsMillis: 30000    # Optional
     numTestsPerEvictionRun: -1              # Optional
     blockWhenExhausted: true                # Optional
     nettyThreads: 32                        # Optional
     subscriptionsPerConnection: 10          # Optional
     subscriptionConnectionPoolSize: 100     # Optional
     pageSizeInBytes: 8192                   # Optional
     readTimeout: 2000                       # Optional
```

### JICS TS-Warteschlangen

```
jics:
   parameters:
     tsqimpl: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   queues:
     ts:
       redis:
         hostName: localhost
         port: 6379
         mode: standalone                      # Optional
         username:                             # Optional
         password: ""                          # Optional
         useSsl: false                         # Optional
```

```
        database: 0                             # Optional
        maxTotal: 128                           # Optional
        maxIdle: 128                            # Optional
        minIdle: 16                             # Optional
        testOnBorrow: true                      # Optional
        testOnReturn: true                      # Optional
        testWhileIdle: true                     # Optional
        testOnCreate: true                      # Optional
        minEvictableIdleTimeMillis: 60000       # Optional
        timeBetweenEvictionRunsMillis: 30000    # Optional
        numTestsPerEvictionRun: -1              # Optional
        blockWhenExhausted: true                # Optional
        nettyThreads: 32                        # Optional
        subscriptionsPerConnection: 10          # Optional
        subscriptionConnectionPoolSize: 100     # Optional
        pageSizeInBytes: 8192                   # Optional
        readTimeout: 2000                       # Optional
```

### Sitzungs-Tracker

```
session-tracker:
   store-type: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                       # Optional
     username:                              # Optional
     password: ""                           # Optional
     useSsl: false                          # Optional
     database: 0                            # Optional
     maxTotal: 128                          # Optional
     maxIdle: 128                           # Optional
     minIdle: 16                            # Optional
     testOnBorrow: true                     # Optional
     testOnReturn: true                     # Optional
     testWhileIdle: true                    # Optional
     testOnCreate: true                     # Optional
     minEvictableIdleTimeMillis: 60000      # Optional
     timeBetweenEvictionRunsMillis: 30000   # Optional
     numTestsPerEvictionRun: -1             # Optional
     blockWhenExhausted: true               # Optional
```

```
    nettyThreads: 32                          # Optional
    subscriptionsPerConnection: 10            # Optional
    subscriptionConnectionPoolSize: 100       # Optional
    pageSizeInBytes: 8192                     # Optional
    readTimeout: 2000                         # Optional
```

### JCL-Checkpoint

```
jcl:
   checkpoint:
     provider: redis
 # If the following redis properties are not specified gapwalk.redis configuration will
 be used for this cache
   redis:
     hostname: localhost
     port: 6379
     mode: standalone                         # Optional
     username:                                # Optional
     password: ""                             # Optional
     useSsl: false                            # Optional
     database: 0                              # Optional
     maxTotal: 128                            # Optional
     maxIdle: 128                             # Optional
     minIdle: 16                              # Optional
     testOnBorrow: true                       # Optional
     testOnReturn: true                       # Optional
     testWhileIdle: true                      # Optional
     testOnCreate: true                       # Optional
     minEvictableIdleTimeMillis: 60000        # Optional
     timeBetweenEvictionRunsMillis: 30000     # Optional
     numTestsPerEvictionRun: -1               # Optional
     blockWhenExhausted: true                 # Optional
     nettyThreads: 32                         # Optional
     subscriptionsPerConnection: 10           # Optional
     subscriptionConnectionPoolSize: 100      # Optional
     pageSizeInBytes: 8192                    # Optional
     readTimeout: 2000                        # Optional
```

### Sperren von Gapwalk-Dateien

```
filesLocks:
   enabled: true
   retryTime: 1000
```

```
  MaxRetry: 5
  provider: redis
# If the following redis properties are not specified gapwalk.redis configuration will
be used for this cache
  redis:
    hostName: localhost
    port: 6379
    mode: standalone                          # Optional
    username:                                 # Optional
    password: ""                              # Optional
    useSsl: false                             # Optional
    database: 0                               # Optional
    pool:
      maxTotal: 128                           # Optional
      maxIdle: 128                            # Optional
      minIdle: 16                             # Optional
      testOnBorrow: true                      # Optional
      testOnReturn: true                      # Optional
      testWhileIdle: true                     # Optional
      testOnCreate: true                      # Optional
      minEvictableIdleTimeMillis: 60000       # Optional
      timeBetweenEvictionRunsMillis: 30000    # Optional
      numTestsPerEvictionRun: -1              # Optional
      blockWhenExhausted: true                # Optional
      nettyThreads: 32                        # Optional
      subscriptionsPerConnection: 10          # Optional
      subscriptionConnectionPoolSize: 100     # Optional
      pageSizeInBytes: 8192                   # Optional
      readTimeout: 2000                       # Optional
```

### Blu4IV-Schlösser

```
blu4iv.lock: redis
blu4iv.lock.timeout: 10 #(in millisecondes)
 # If the following redis properties are not specified gapwalk.redis configuration
will be used for this cache
blu4iv.lock.redis:
      hostName: localhost
      port: 6379
      mode: standalone                        # Optional
      username:                               # Optional
      password: ""                            # Optional
```

```
      useSsl: false                          # Optional
      database: 0                            # Optional
      maxTotal: 128                          # Optional
      maxIdle: 128                           # Optional
      minIdle: 16                            # Optional
      testOnBorrow: true                     # Optional
      testOnReturn: true                     # Optional
      testWhileIdle: true                    # Optional
      testOnCreate: true                     # Optional
      minEvictableIdleTimeMillis: 60000      # Optional
      timeBetweenEvictionRunsMillis: 30000   # Optional
      numTestsPerEvictionRun: -1             # Optional
      blockWhenExhausted: true               # Optional
      nettyThreads: 32                       # Optional
      subscriptionsPerConnection: 10         # Optional
      subscriptionConnectionPoolSize: 100    # Optional
      pageSizeInBytes: 8192                  # Optional
      readTimeout: 2000                      # Optional
```

### Datensatz-Katalog

```
datasimplifier:
   catalogImplementation: redis
   # If the following redis properties are not specified gapwalk.redis configuration
 will be used for this cache
   redis:
     hostName: localhost
     port: 6379
     mode: standalone                       # Optional
     username:                              # Optional
     password: ""                           # Optional
     useSsl: false                          # Optional
     database: 0                            # Optional
     maxTotal: 128                          # Optional
     maxIdle: 128                           # Optional
     minIdle: 16                            # Optional
     testOnBorrow: true                     # Optional
     testOnReturn: true                     # Optional
     testWhileIdle: true                    # Optional
     testOnCreate: true                     # Optional
     minEvictableIdleTimeMillis: 60000      # Optional
     timeBetweenEvictionRunsMillis: 30000   # Optional
     numTestsPerEvictionRun: -1             # Optional
```

```
    blockWhenExhausted: true                # Optional
    nettyThreads: 32                        # Optional
    subscriptionsPerConnection: 10          # Optional
    subscriptionConnectionPoolSize: 100     # Optional
    pageSizeInBytes: 8192                   # Optional
    readTimeout: 2000                       # Optional
```

### Geheimer Manager für Redis-Caches

Die `application-main.yaml` Datei kann auf den geheimen ARN für Redis-Caches verweisen. Informationen zur Integration AWS Secrets Manager zum sicheren Abrufen von Redis-Verbindungsdetails zur Laufzeit finden Sie unter. the section called “AWS Die Geheimnisse von Blu Age Runtime”

## Konfigurieren Sie die 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


- URI-Zugänglichkeit für Gapwalk-Anwendungen konfigurieren
- Konfigurieren Sie die Authentifizierung für Gapwalk-Anwendungen


### URI-Zugänglichkeit für Gapwalk-Anwendungen konfigurieren

In diesem Thema wird beschrieben, wie Sie die Filterung URIs für Gapwalk-Anwendungen konfigurieren. Für diese Funktion ist kein Identitätsanbieter (IdP) erforderlich.

Um eine Liste von zu blockieren URIs, fügen Sie `application-main.yml` der Ihrer modernisierten Anwendung die folgenden beiden Zeilen hinzu und ersetzen Sie *`URI-1`* dabei*`URI-2`*, usw., durch die Zeile, 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

Um die OAuth2 Authentifizierung für Ihre Gapwalk-Anwendung zu konfigurieren, müssen Sie einen Identitätsanbieter (IdP) einrichten und ihn in Ihre Anwendung integrieren. Dieses Handbuch behandelt die Schritte zur Verwendung von Amazon Cognito oder Keycloak als IdP. Mit Amazon Cognito können Sie die Konfigurationsdatei Ihrer Anwendung mit den Cognito-Benutzerpooldetails aktualisieren. Mit Keycloak können Sie den Zugriff auf Ihre Anwendung APIs und Ressourcen auf der Grundlage der dem Benutzer zugewiesenen Rollen steuern.

Themen


- OAuth2 Gapwalk-Authentifizierung mit Amazon Cognito konfigurieren
- Konfigurieren Sie die Gapwalk-Authentifizierung OAuth2 mit Keycloak


### OAuth2 Gapwalk-Authentifizierung mit Amazon Cognito konfigurieren

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.

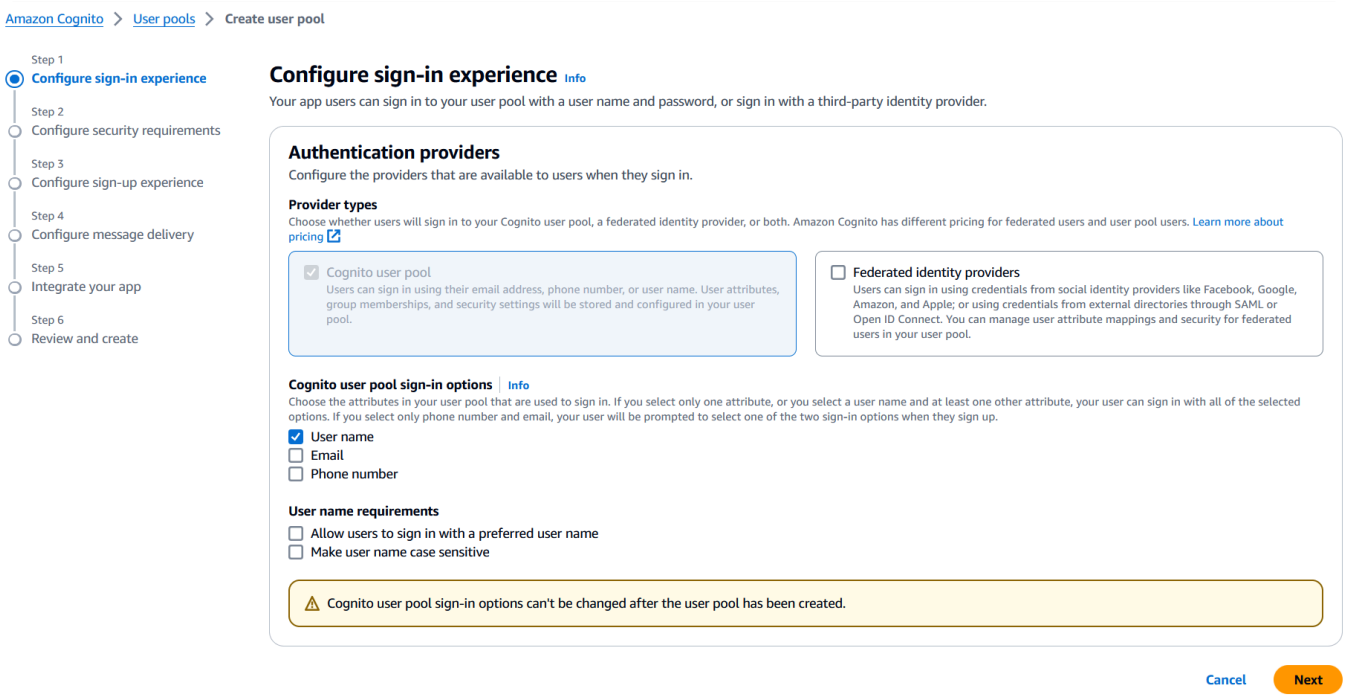
### Einrichtung von Amazon Cognito

Zunächst erstellen und konfigurieren wir einen Amazon Cognito Cognito-Benutzerpool und einen Benutzer, die wir mit unserer bereitgestellten Gapwalk-Anwendung zu Testzwecken verwenden werden.

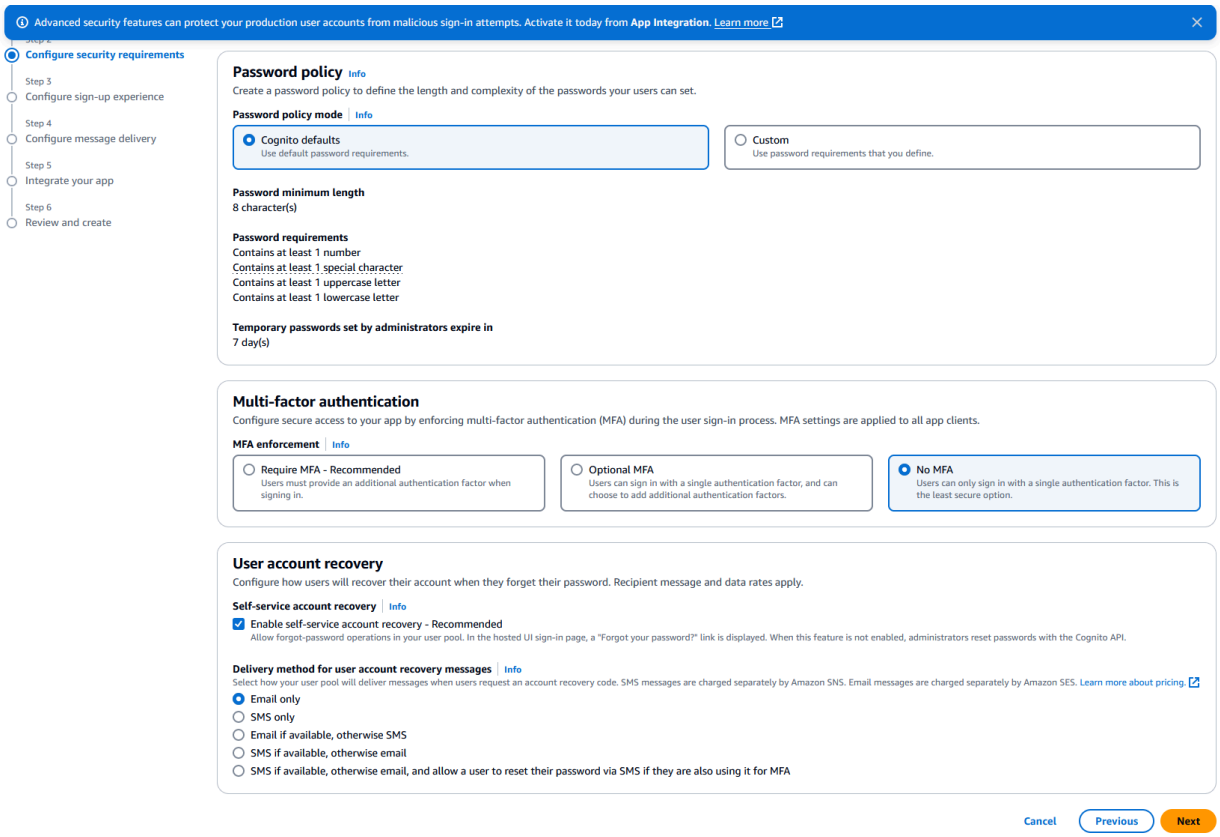
> **Note**
>
> Wenn Sie einen anderen IdP verwenden, können Sie diesen Schritt überspringen.

#### Benutzerpool erstellen

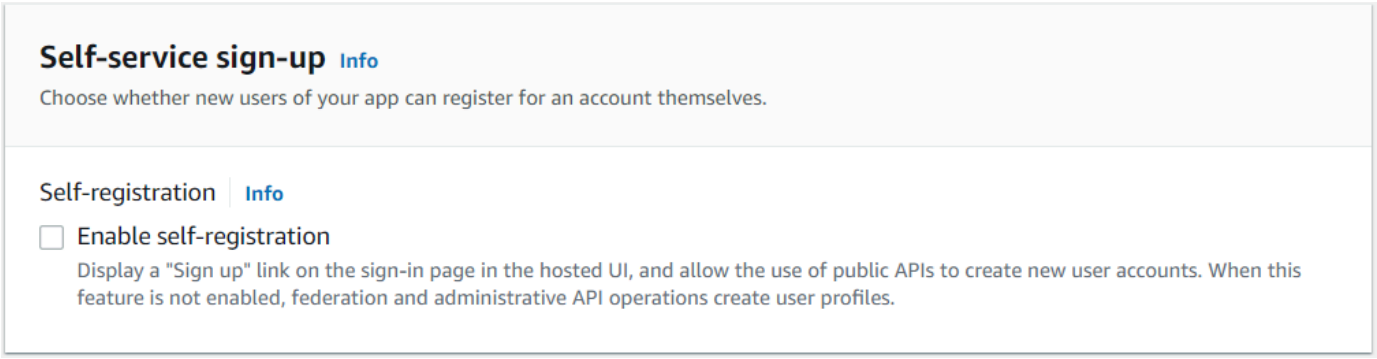
1. Gehen Sie zu Amazon Cognito in AWS Management Console und authentifizieren Sie sich mit Ihren AWS Anmeldeinformationen.
2. Wählen Sie User Pools (Benutzerpools) aus.
3. Wählen Sie Create a user pool.
4. Behalten Sie unter Anmeldeerlebnis konfigurieren den Standardanbietertyp für den Cognito-Benutzerpool bei. Sie können eine oder mehrere Anmeldeoptionen für den Cognito-Benutzerpool auswählen. Wählen Sie zunächst Benutzername und dann Weiter.



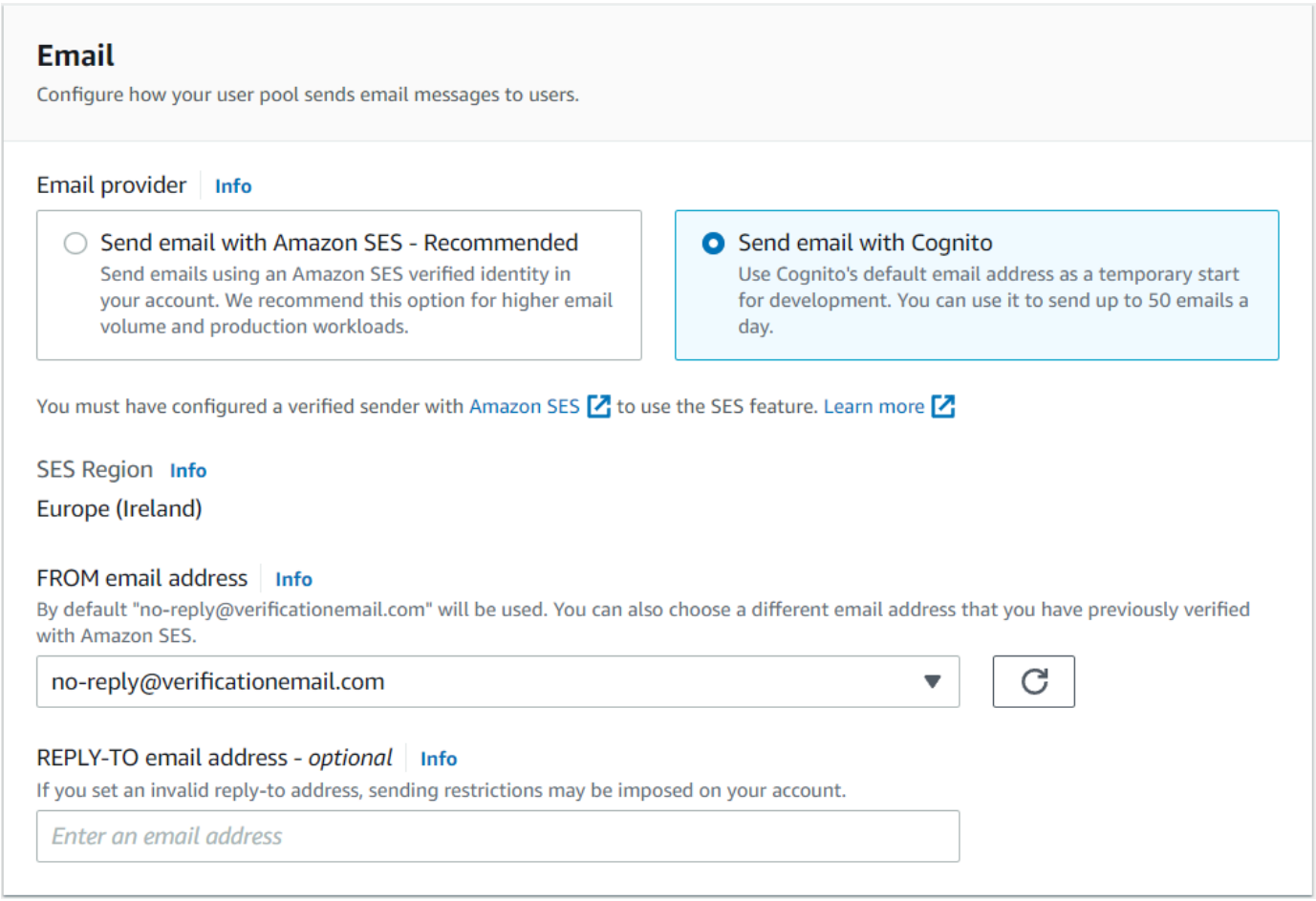

5. Behalten Sie unter Sicherheitsanforderungen konfigurieren die Standardeinstellungen bei und deaktivieren Sie die Multi-Faktor-Authentifizierung, indem Sie Kein MFA und dann Weiter auswählen.



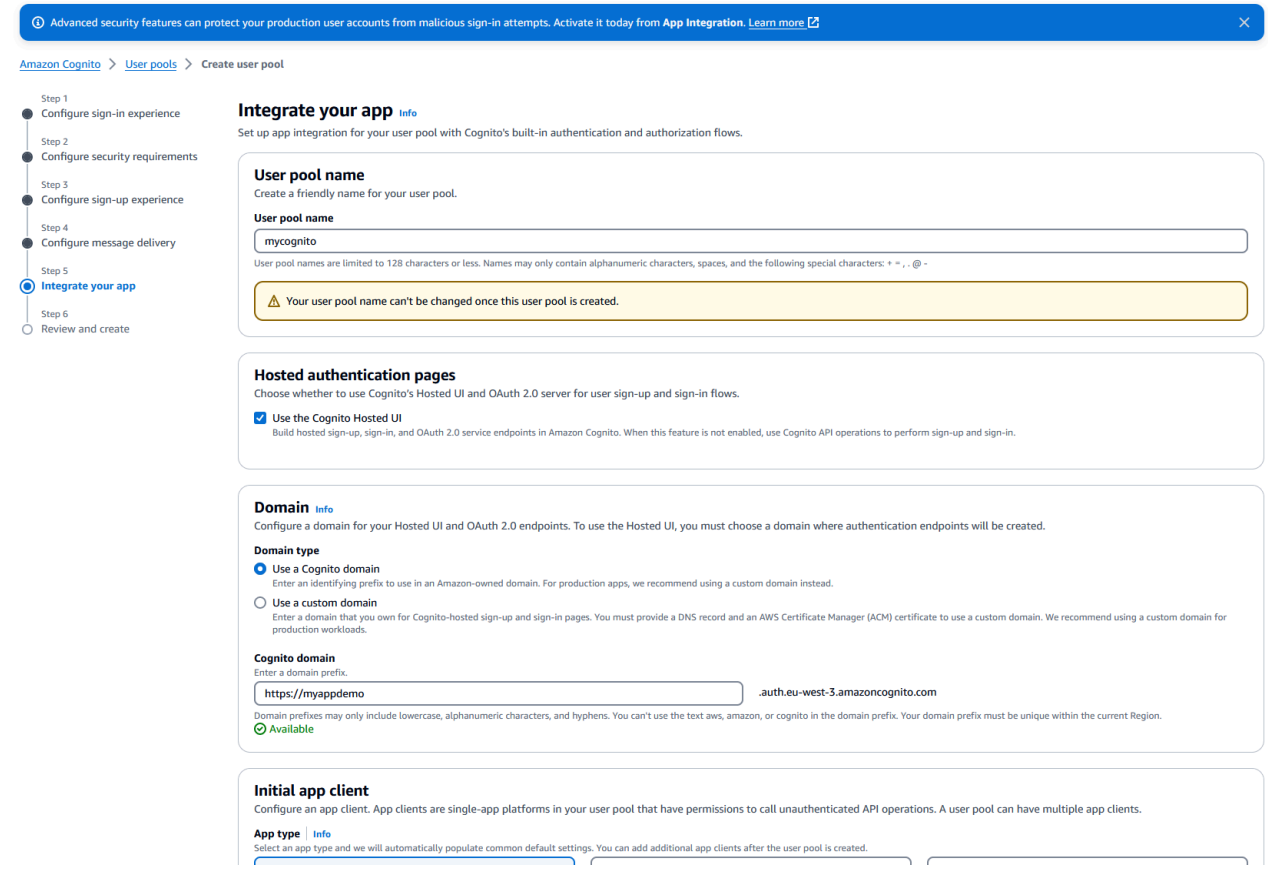
6. Deaktivieren Sie aus Sicherheitsgründen die Option Selbstregistrierung aktivieren und wählen Sie dann Weiter aus.




7. Wählen Sie E-Mail mit Cognito senden und dann Weiter.




8. Geben Sie unter Integrieren Sie Ihre App einen Namen für Ihren Benutzerpool an. Wählen Sie unter Gehostete Authentifizierungsseiten die Option Die von Cognito gehostete Benutzeroberfläche verwenden aus.



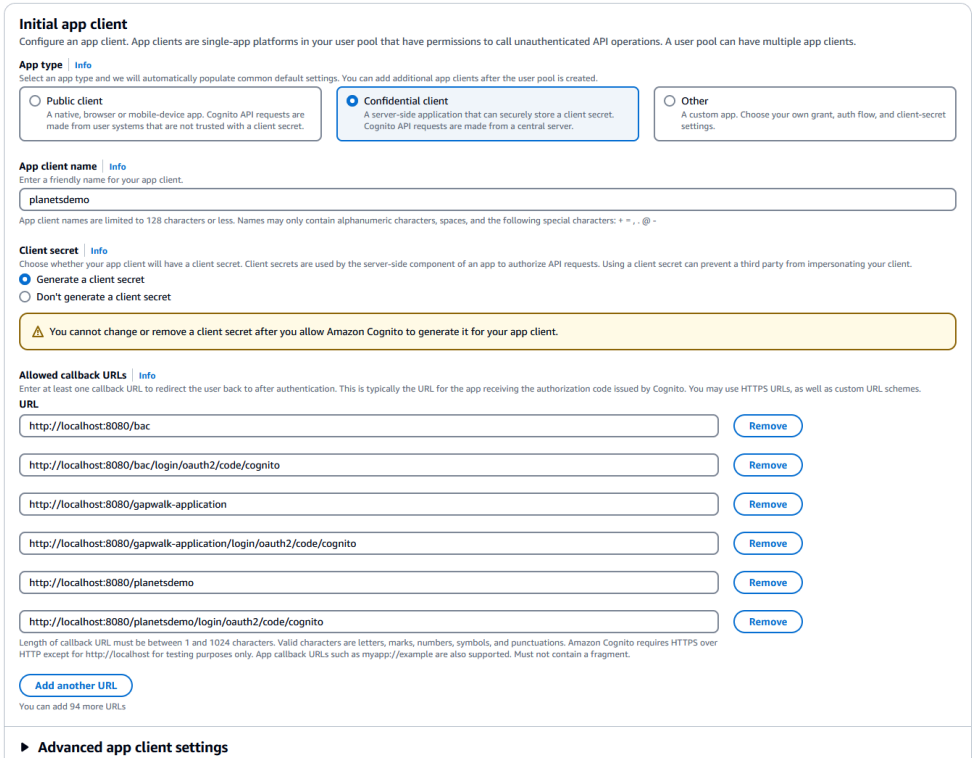

9. Der Einfachheit halber wählen Sie unter Domain die Option Eine Cognito-Domäne verwenden aus und geben Sie ein Domainpräfix ein, `https://planetsdemo` z. B. Die Demo-App muss als Client hinzugefügt werden.

   a. 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.

   b. 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.




c. URLsGeben Sie unter Zulässige Abmeldung 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.

d. Behalten Sie die Standardwerte in den Abschnitten Erweiterte App-Client-Einstellungen und Lese- und Schreibberechtigungen für Attribute bei.

e. Wählen Sie Weiter aus.

10. 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 wählen Sie dann unter Benutzer die Option Benutzer erstellen aus.

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.

Integrieren Sie 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
            jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/
jwks.json
            token-uri: ${gapwalk-application.security.domainName}/oauth2/token
            user-name-attribute: username
      resourceserver:
        jwt:
          jwk-set-uri: ${gapwalk-application.security.issuerUri}/.well-known/jwks.json
```

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`* Benutzerpool-ID.
3. Wählen Sie dort App-Integration aus, wo Sie Ihre finden können*`your-cognito-domain`*, und gehen Sie dann zu App-Clients und Analysen und wählen Sie Ihre App aus.
4. Im App-Client: YourApp findest du*`client-name`*,*`client-id`*, und *`client-secret`* (Geheimes 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. *`redirect-uri`*Geben Sie den URI ein, den Sie für Zulässige Rückruf-URL angegeben haben. In unserem Beispiel ist `http://localhost:8080/planetsdemo/login/oauth2/code/cognito` es.

Sie können jetzt Ihre Gapwalk-Anwendung bereitstellen und den zuvor erstellten Benutzer verwenden, um sich bei Ihrer App anzumelden.

Konfigurieren Sie die Gapwalk-Authentifizierung OAuth2 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.

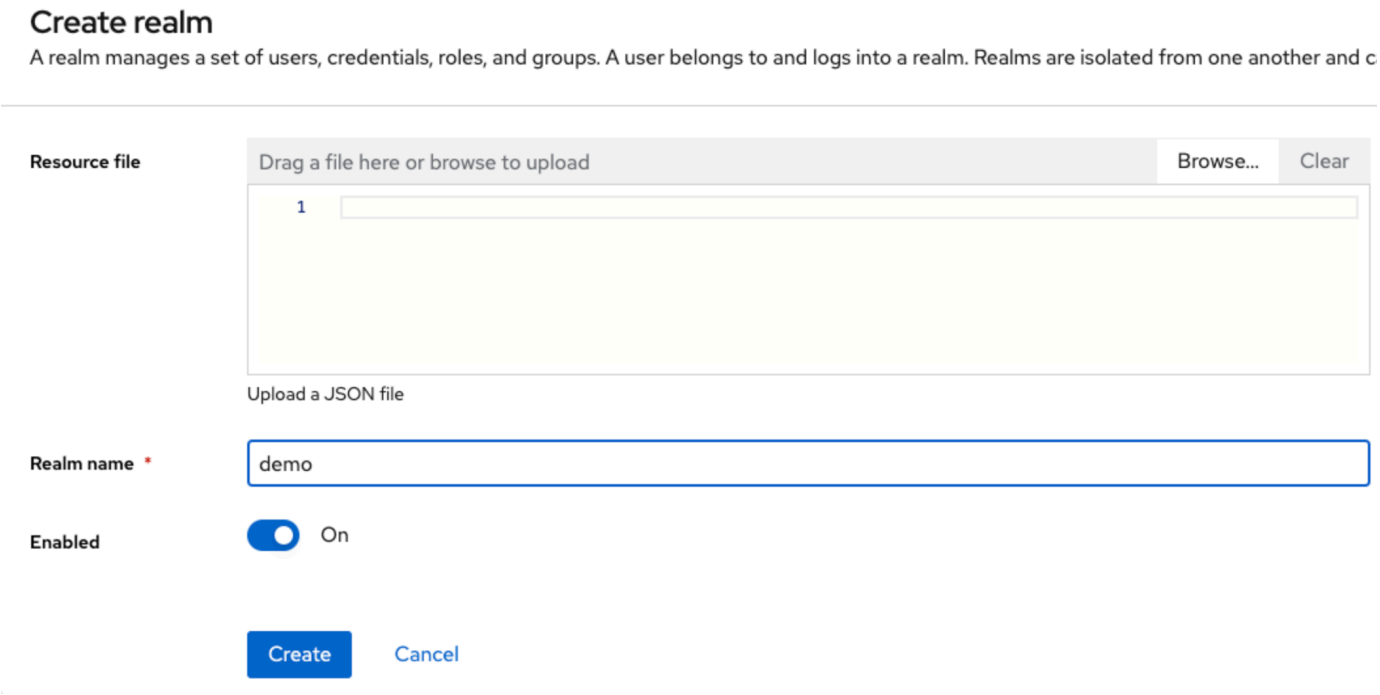
Voraussetzungen

- Keycloak

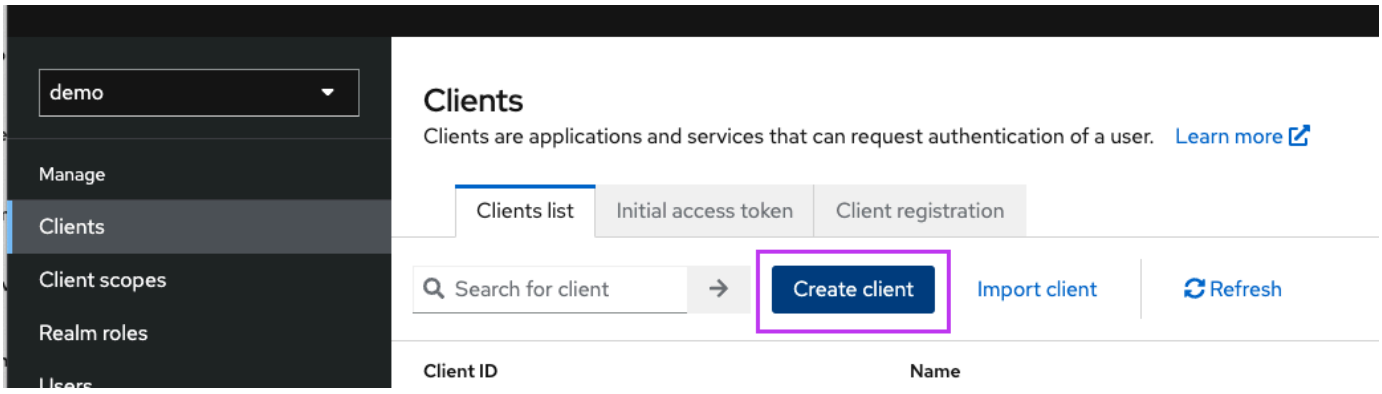
- Gapwalk-Anwendung

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.




2. Erstellen Sie einen Client mit dem Namen**`app-demo`**.




`localhost:8080`Ersetzen Sie durch die Adresse Ihrer Gapwalk-Anwendung

## General settings

**Client ID** * app-demo

**Name**

**Description**

**Always display in UI** Off

## Access settings

**Root URL** http://localhost:8080

**Home URL**

**Valid redirect URIs**
- http://localhost:8080/*
- https://localhost:8080/*

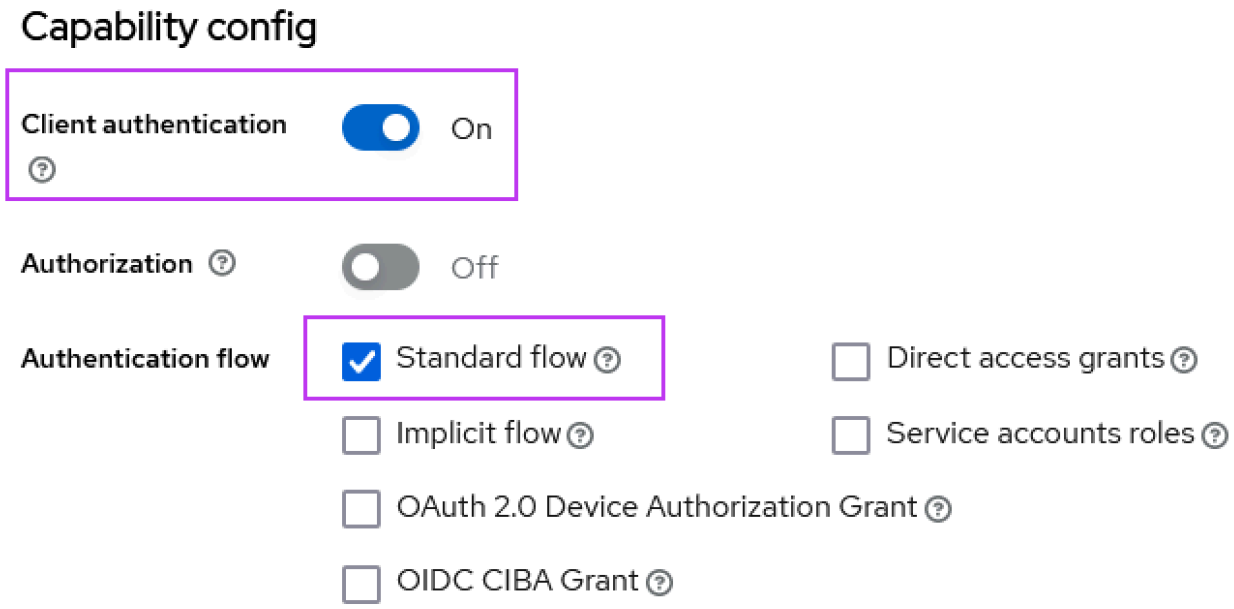
Add valid redirect URIs

**Valid post logout redirect URIs**
- http://localhost:8080/*
- https://localhost:8080/*

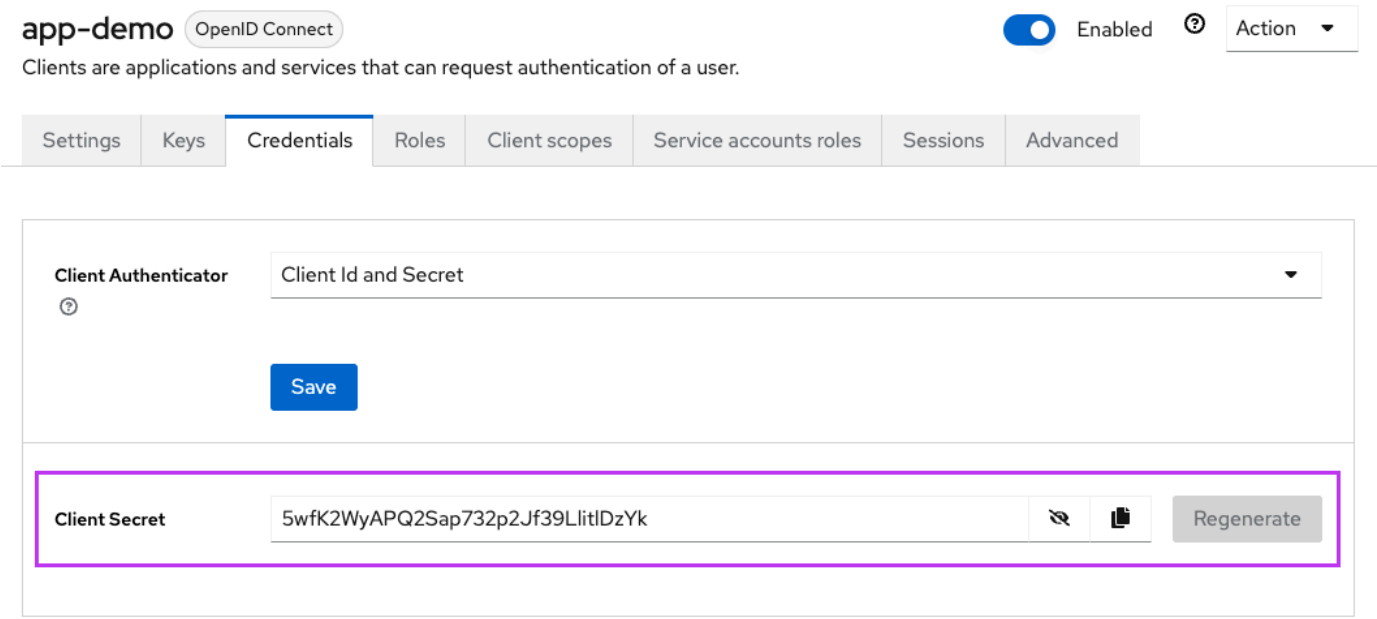
Add valid post logout redirect URIs

**Web origins**
- +

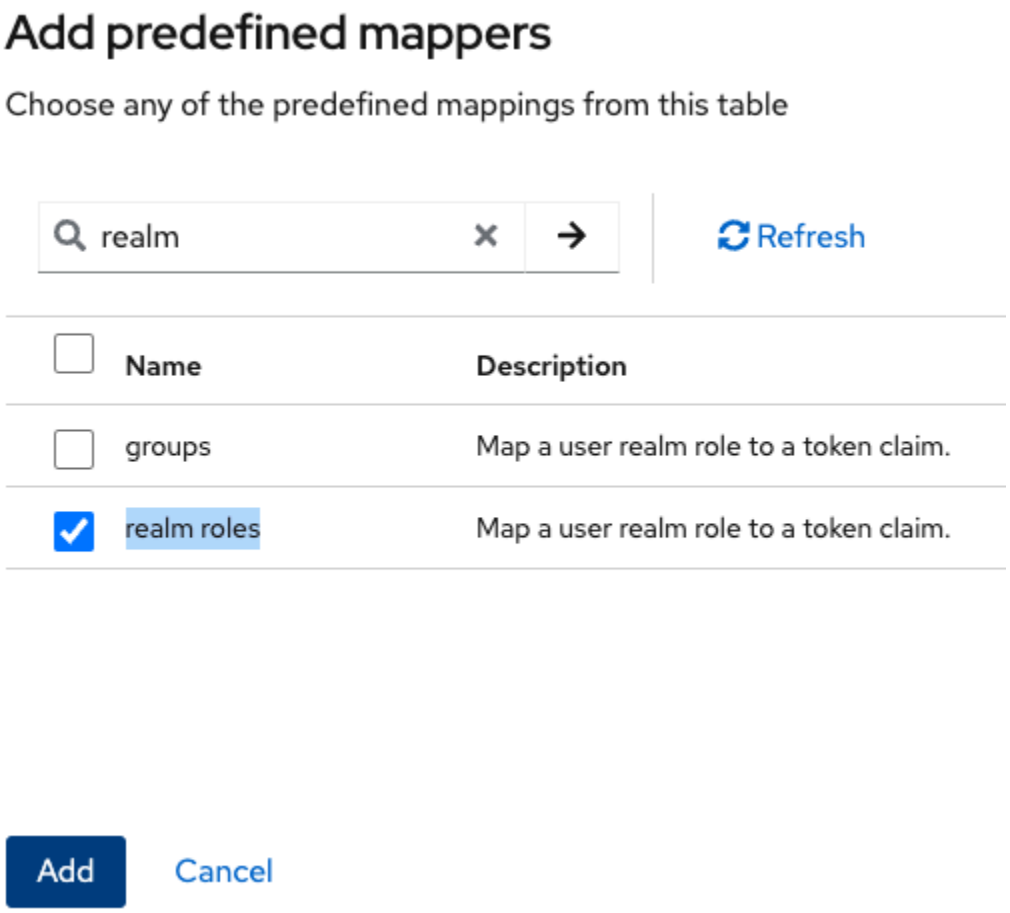
Add web origins



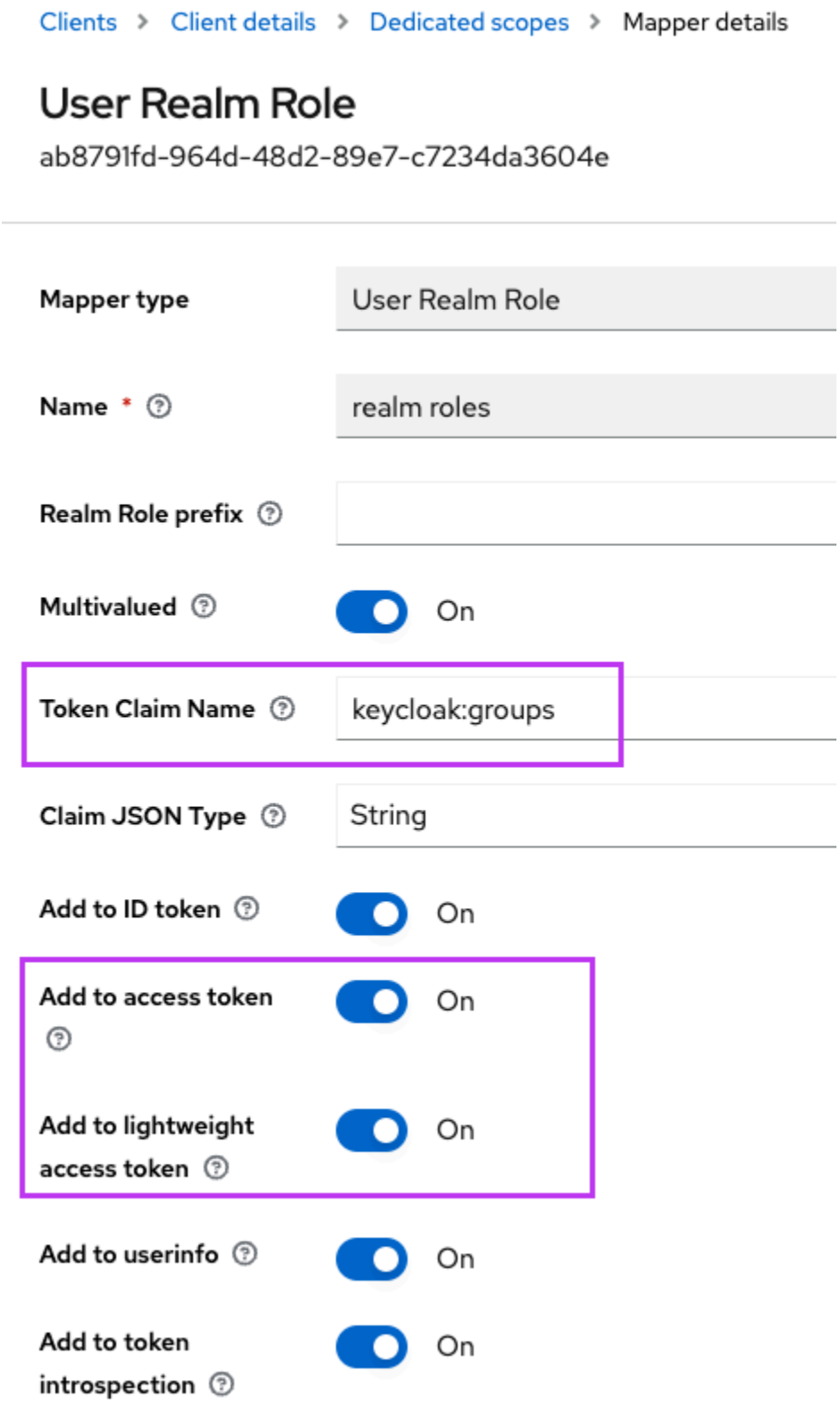

3. Um Ihr Client-Geheimnis zu erhalten, wählen Sie Clients, dann App-Demo und dann Credentials.



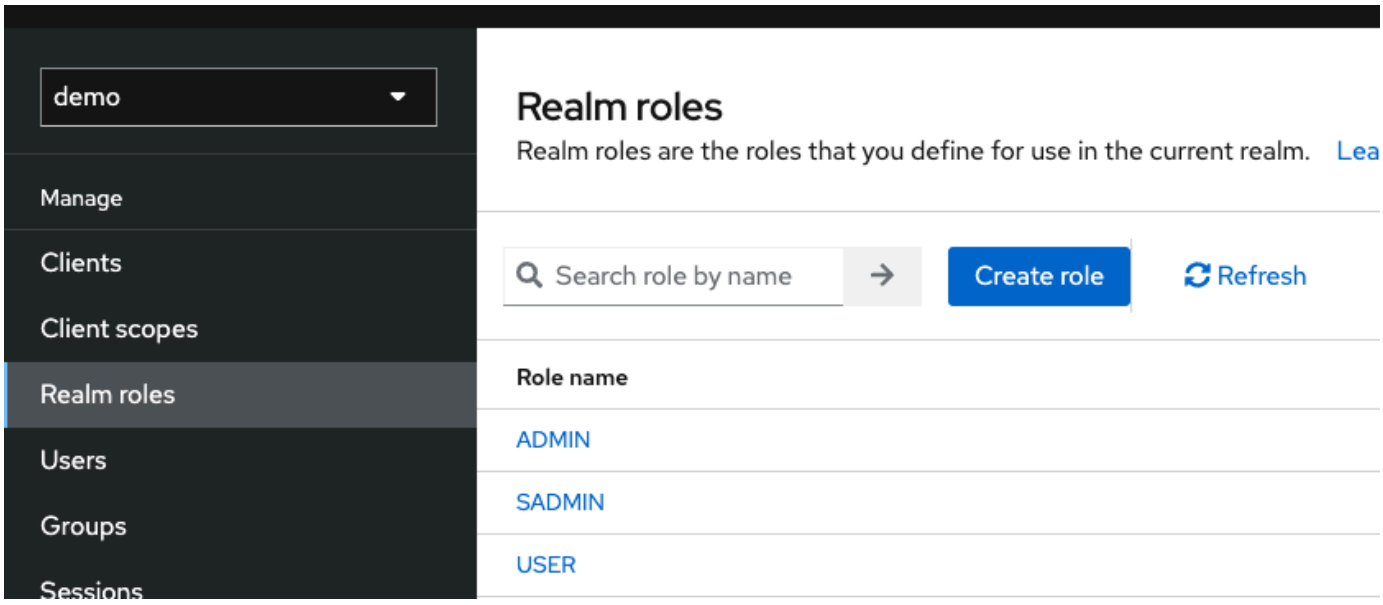

4. Wählen Sie Clients, dann Client-Bereiche und dann Vordefinierten Mapper hinzufügen aus. Wählen Sie Realm-Rollen aus.



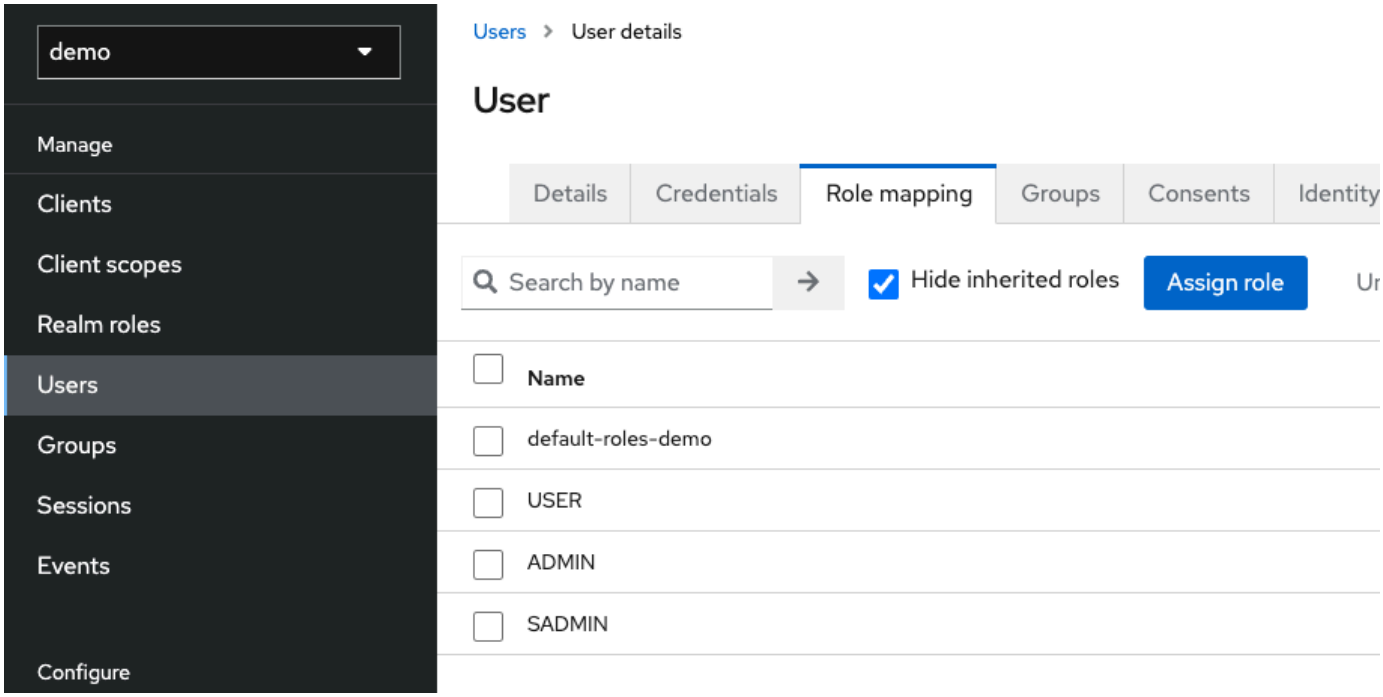

5. Bearbeiten Sie Ihre Realm-Rolle mit der in der folgenden Abbildung gezeigten Konfiguration.


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
On
Token Claim Name
keycloak:groups
Claim JSON Type
String
Add to ID token
On
Add to access token
On
Add to lightweight access token
On
Add to userinfo
On
Add to token introspection
On

6. 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.




7. Wählen Sie Realms-Rollen und erstellen Sie 3 Rollen:**`SUPER_ADMIN`**, **`ADMIN`** und. **`USER`** Diese Rollen werden später der Gapwalk-Anwendung `ROLE_SUPER_ADMINROLE_ADMIN`, und `ROLE_USER` zugeordnet, um auf einige eingeschränkte API-REST-Aufrufe zugreifen zu können.



### Integrieren Sie 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 Runtime 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

- Verfügbare Endpunkte für Benutzer beim Erstellen URLs
- Endpunkte für die Gapwalk-Anwendung in Blu Age AWS
- REST-Endpunkte der Blusam-Anwendungskonsole
- JICS-Anwendungskonsole in AWS Blu Age verwalten
- Datenstrukturen für AWS Blu Age-Benutzer

## Verfügbare Endpunkte für Benutzer beim Erstellen URLs

In diesem Thema werden die URLs Root-Pfade für Endgeräte aufgeführt. 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.

## Endpunkte für die Gapwalk-Anwendung in Blu Age AWS

In diesem Thema erfahren Sie mehr über die Endpunkte für die Gapwalk-Webanwendung. Diese verwenden den Stammpfad. `/gapwalk-application`

Themen


- Endgeräte im Zusammenhang mit Batch-Jobs (modernisiert JCLs und ähnlich)
- Endpunkte von Metriken
- Andere Endpunkte
- Endpunkte im Zusammenhang mit Jobwarteschlangen


### Endgeräte im Zusammenhang mit Batch-Jobs (modernisiert JCLs und ähnlich)

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).

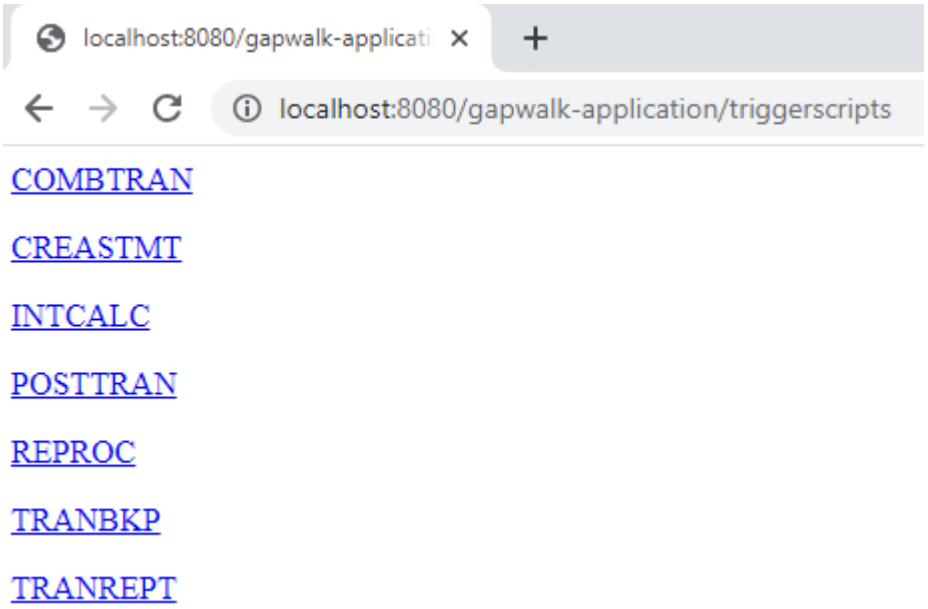
Beispielantwort:

```
<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 bezeichnet`Map<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 a bezeichnet`Map<String,String>`) an das Skript übergeben werden. Die angegebenen Parameter werden automatisch zu den https://docs.groovy-lang.org/latest/html/api/groovy/lang/Binding.html [Bindungen] 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 von einer festen, begrenzten Anzahl von Threads abhängt, 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 Details zur Auftragsausführung, Nachrichtenstruktur.

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 eine JSON-Zeichenfolge zurück, die einzelne Details zur Jobausführung darstellt (siehe Details zur Auftragsausführung, Nachrichtenstruktur) 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/{identifier:.+}`
- Argument: ID für die Auftragsausführung (verpflichtend): Die eindeutige Kennung für die Auftragsausführung, die gewaltsam beendet werden soll.
- Gibt eine Textnachricht zurück, 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 Vorgängerschritte oberhalb des Checkpoints ausführen zu müssen).

- Unterstützte Methode: GET
- Pfad: `/restarts/{scriptId}/{jobId}`
- Argumente:
  - ScriptID (optional — Zeichenfolge): Das Skript wird neu gestartet.
  - jobId (optional — string): Die eindeutige Kennung einer Jobausführung.

- Gibt eine JSON-formatierte Liste vorhandener Neustartpunkte zurück, die verwendet werden können, um einen Job neu zu starten, dessen Ausführung nicht ordnungsgemäß abgeschlossen wurde, oder um einen verzögerten Neustart auszulösen, indem zuvor ausgeführte Schritte umgangen werden. Wenn keine Checkpoints durch Skripts registriert wurden, lautet der Seiteninhalt „Keine registrierten Checkpoints“.

Einen Job neu starten (synchron)

- Unterstützte Methode: GET
- Pfad: `/restart/{hashcode}/{scriptId}/{skipflag}`
- Argumente:
  - Hashcode (Ganzzahl — verpflichtend): Startet die letzte Ausführung eines Jobs 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).
  - ScriptID (optional — Zeichenfolge): Das Skript wird neu gestartet.
  - skipflag (optional — boolean): Überspringt die Ausführung des ausgewählten Schritts (Checkpoint) und führt einen Neustart vom unmittelbaren Nachfolgeschritt aus (falls vorhanden).
- Rücksendungen: siehe Beschreibung der Rücksendung oben. `/script`

Einen Job neu starten (asynchron)

- Unterstützte Methode: GET
- Pfad: `/triggerrestart/{hashcode}/{scriptId}/{skipflag}`
- Argumente:
  - Hashcode (Ganzzahl — verpflichtend): Startet die letzte Ausführung eines Jobs 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).
  - ScriptID (optional — Zeichenfolge): Das Skript wird neu gestartet.
  - skipflag (optional — boolean): Überspringt die Ausführung des ausgewählten Schritts (Checkpoint) und führt einen Neustart vom unmittelbaren Nachfolgeschritt aus (falls vorhanden).
- 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 entsprechend den Hostkapazitäten (Anzahl 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).

### Informationen zur Auftragsausführung löschen

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

## Endpunkte von Metriken

### JVM

Dieser Endpunkt gibt verfügbare Metriken zurück, die sich auf die JVM beziehen.

- 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.
  - numberOfElements: 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 Kennzahlen zur Transaktionsausführung.
  - pageNumber: Aktuelle Seitennummer bei der Paginierung.
  - pageSize: Anzahl der pro Seite angezeigten Elemente.
  - totalPages: Gesamtzahl der verfügbaren Seiten.
  - numberOfElements: 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)
- Listet aktive Sitzungen auf


### 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 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 eine `com.netfective.bluage.gapwalk.rt.shared.web.TransactionResultBean` JSON-Serialisierung zurück, 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 eine `com.netfective.bluage.gapwalk.rt.shared.web.RecordHolderBean` JSON-Serialisierung zurück, die das Ergebnis des Starts der JICS-Transaktion darstellt. Die Einzelheiten der Struktur finden Sie in. Aufbau des Transaktionsstarts, Aufzeichnung der Ergebnisse

Listet aktive Sitzungen auf

- unterstützte Methoden: GET, POST
- Pfad: `/activesessionlist`
- Argumente: keine
- Gibt eine Liste von `com.netfective.bluage.gapwalk.application.web.sessiontracker.SessionTrackerObje`

InJSON-Serialisierungen zurück, die die Liste der aktiven Benutzersitzungen darstellt. Wenn die Sitzungsverfolgung deaktiviert ist, wird eine leere Liste zurückgegeben.

## Endpunkte im Zusammenhang mit Jobwarteschlangen

Jobwarteschlangen sind die AWS Blu-Age-Unterstützung für den Mechanismus zur Einreichung von AS4 00-Jobs. Jobwarteschlangen werden in AS4 00 verwendet, um Jobs in bestimmten Threadpools auszuführen. Eine Auftragswarteschlange 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 unterMöglicher 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 eine JSON-Datei zurück, die das Original `SubmitJobMessage` und ein Protokoll enthält, 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 eine Liste aller geplanten Jobs als JSON-Zeichenfolge zurück. Eine Beispielantwort finden Sie unterListe der Antworten auf geplante Jobs.

Geben Sie alle Jobs frei, die sich in der Warteschleife befinden

- Unterstützte Methode: POST
- Pfad: `/release-all`
- Gibt eine Meldung zurück, die das Ergebnis des Freigabeversuchs 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. Es ist ein unbekannter Fehler 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

Für einen bestimmten Jobnamen können mehrere Jobs mit unterschiedlichen Job-Nummern eingereicht werden (die Einheitlichkeit eines ausgeführten Jobs wird von mehreren vergeben <job name, job number>). Der Endpunkt versucht, alle Jobübermittlungen mit dem angegebenen Jobnamen, die sich im Status „in der Warteschleife“ befinden, freizugeben.

- Unterstützte Methode: POST
- Pfad: `/release/{name}`
- Argumente: Der Jobname, nach dem gesucht werden soll, als Zeichenfolge. Zwingend erforderlich.
- Gibt eine Meldung zurück, 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 sich für das angegebene Paar in der Warteschleife befindet <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.

- <name/number>HTTP 503 und die Meldung „Job>Not released“. 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.netfective.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 eine Liste aller geplanten Jobs als JSON-Zeichenfolge zurück.

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-Anwendungskonsole

In diesem Abschnitt erfahren Sie mehr über die Blusam-Anwendungskonsole, eine API, die die Verwaltung modernisierter VSAM-Datensätze vereinfachen soll. Endpunkte für die Blusam-Webanwendung verwenden den Root-Pfad. `/bac`

Themen


- Datensätze und zugehörige Endpunkte
- Massendatensätze, verwandte Endpunkte
- Datensätze
- Masks
- Sonstige
- BAC-Endpunkte für die Benutzerverwaltung


## 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.

  Cache aktivieren

  (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 mit der hochgeladenen Datei verknüpft 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

Beispielanforderung:

```
/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.

  kmsKeyId

  (optional, Zeichenfolge): Die AWS KMS ID, die für die Verschlüsselung des exportierten Datensatzes in den Amazon S3 S3-Bucket verwendet werden soll.

- Gibt den exportierten Datensatz zurück

Beispielanforderung:

```
/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.

datasetOutputFile

(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, BACRead Parameter): 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.

Cache aktivieren

(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 Folgendes 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:

  paramFilePath

  (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`
- Pfadvariablen:

  Datensatzgröße:... /loadMasks/ {RecordSize}

  (optional, numerisch): Die Datensatzgröße, filtern Sie geladene Masken, die dieser Datensatzgröße entsprechen
- 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: POST
- 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 Persistenz, die als Zeichenfolge verwendet wird

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.

## BAC-Endpunkte für die Benutzerverwaltung

Verwenden Sie die folgenden Endpunkte, um Benutzerinteraktionen zu verwalten.

Themen

- Melden Sie einen Benutzer an
- Überprüfen Sie, ob mindestens ein Benutzer im System vorhanden ist
- Einen neuen Benutzer aufnehmen
- Benutzerinformationen abrufen
- Auflisten von Benutzern
- Löschen eines Benutzers
- Meldet den aktuellen Benutzer ab

Melden Sie einen Benutzer an

- Unterstützte Methode: POST
- Pfad: `/api/services/security/servicelogin/login`
- Argumente: Keine
- Gibt die JSON-Serialisierung eines `com.netfective.bluage.bac.entities.SignOn` Objekts zurück, die den Benutzer darstellt, dessen Anmeldeinformationen in der aktuellen Anfrage angegeben wurden. Das Passwort ist im zurückgegebenen Objekt nicht sichtbar. Die Rollen, die dem Benutzer zugewiesen wurden, werden aufgelistet.

Beispielantwort:

```
{
     "login": "some-admin",
     "password": null,
     "roles": [
       {
         "id": 0,
         "roleName": "ROLE_ADMIN"
       }
     ]
   }
```

Überprüfen Sie, ob mindestens ein Benutzer im System vorhanden ist

- 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 aufnehmen

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die Rolle ROLE_ADMIN.
- Pfad: `/api/services/security/servicelogin/recorduser`
- Argumente: Die JSON-Serialisierung eines `com.netfective.bluage.bac.entities.SignOn` Objekts, das den Benutzer darstellt, der dem Speicher hinzugefügt werden soll. Die Rollen für den Benutzer müssen definiert werden, andernfalls kann der Benutzer die BAC-Einrichtung und die Endpunkte möglicherweise nicht verwenden.
- Gibt den booleschen Wert zurück, `true` wenn der Benutzer erfolgreich erstellt wurde. Gibt andernfalls zurück`false`.
- JSON-Beispiel für eine Anfrage:

```
{
    "login": "simpleuser",
    "password": "simplepassword",
    "roles": [
      {
        "id": 2,
        "roleName": "ROLE_USER"
      }
    ]
  }
```

  Im Folgenden sind die beiden gültigen Werte für aufgeführt`roleName`:

  - `ROLE_ADMIN`: kann Blusam-Ressourcen und Benutzer verwalten.
  - `ROLE_USER`: kann Blusam-Ressourcen verwalten, aber keine Benutzer.

Benutzerinformationen abrufen

- Unterstützte Methode: GET
- Pfad: `/api/services/security/servicelogin/userInfo`

- Argumente: Keine
- Gibt den Benutzernamen und die Rolle des aktuell verbundenen Benutzers zurück

Auflisten von Benutzern

- Unterstützte Methode: GET
- Erfordert Authentifizierung und die Rolle ROLE_ADMIN.
- Pfad: `/api/services/security/servicelogin/listusers`
- Argumente: Keine
- Gibt eine Liste von zurück`com.netfective.bluage.bac.entities.SignOn`, serialisiert als JSON.

Löschen eines Benutzers

> **Important**
>
> Diese Aktion kann nicht rückgängig gemacht werden. Der gelöschte Benutzer kann sich nicht wieder mit der BAC-Anwendung verbinden.

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die Rolle ROLE_ADMIN.
- Pfad: `/api/services/security/servicelogin/deleteuser`
- Argumente: Die JSON-Serialisierung eines `com.netfective.bluage.bac.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.

Meldet 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.

# JICS-Anwendungskonsole in AWS Blu Age verwalten

Die JICS-Komponente ist die AWS Blu-Age-Unterstützung für die Modernisierung der älteren CICS-Ressourcen. Die Webanwendung der JICS-Anwendungskonsole 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 der JICS-Anwendungskonsole 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)
- Bearbeiten der eigenen GRUPPEN mehrerer LISTEN
- Löscht eine LISTE
- Löscht eine GRUPPE
- Löschen Sie eine TRANSAKTION
- Löschen Sie 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
- Aktualisieren Sie eine LISTE
- 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 verärgern
- 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 bereitzustellen. Weitere Informationen zur Organisation der CICS-Ressourcen finden Sie unter CICS-Ressourcen.

- Unterstützte Methode: GET
- Erfordert Authentifizierung und eine der 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 eine der 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 eine der folgenden Rollen: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Pfad: `/api/services/rest/jicsservice/listJicsGroups`
- Argumente: Keine
- Gibt eine Liste serialisierter JicsContainer Objekte (GROUPS) als JSON zurück. 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 eine der 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 Sie suchen. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.JACList`

  Beispielanfrage:

```
{
    "jacType":"JACList",
    "name":"MURACHS",
    "isActive":true
  }
```

- Gibt eine Liste von serialisierten JicsContainer Objekten (GROUPS) als JSON zurück, 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 eine der 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 Sie suchen. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.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 eine Liste serialisierter JicsResource Objekte zurück, die der angegebenen GRUPPE gehören. Die Objekte werden in keiner bestimmten Reihenfolge zurückgegeben und sind unterschiedlichen Typs, wie PROGRAM, TRANSACTION, FILE usw.

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 Sie 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, wie PROGRAM, TRANSACTION, FILE usw.

Bearbeiten der eigenen GRUPPEN mehrerer LISTEN

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der folgenden Rollen: ROLE_ADMIN, ROLE_SUPER_ADMIN, ROLE_USER
- Pfad: `/api/services/rest/jicsservice/editGroupsList`

- Argumente: eine JSON-Darstellung 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“ Sie 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 LISTS-Änderungen erfolgreich im zugrunde liegenden JICS-Speicher gespeichert.

Löscht eine LISTE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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, wurde die LIST-Löschung erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Löscht eine GRUPPE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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, wurde die GROUP-Löschung erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Löschen Sie eine TRANSAKTION

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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.netfective.bluage.jac.entities.JACTransaction`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TRANSACTION-Löschung erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Löschen Sie ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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.netfective.bluage.jac.entities.JACProgram`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das Löschen von PROGRAMM erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Löscht eine DATEI

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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.netfective.bluage.jac.entities.JACFile`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das Löschen der DATEI im zugrunde liegenden JICS-Speicher erfolgreich durchgeführt.

Löscht eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 einer `com.netfective.bluage.jac.entities. JACTDQueue`Objekt.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TDQUEUE-Löschung erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Löscht ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 einer `com.netfective.bluage.jac.entities. JACTSModel`Objekt.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TSMODEL-Löschung erfolgreich auf dem zugrunde liegenden JICS-Speicher ausgeführt.

Elemente löschen

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 eine der 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 einer `com.netfective.bluage.jac.entities. JACList`Objekt.
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die LISTE erfolgreich 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-Payload, die die zu erstellende 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, 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: Sie versuchen, die Transaktion 'ARIT' in der Gruppe 'TEST' zu erstellen, aber eine Transaktion mit diesem Namen ist in dieser Gruppe bereits vorhanden.

```
{
    "jacType":"JACTransaction",
    "name":"ARIT",
    "groupName":"TEST",
    "isActive":true
  }
```

Sie erhalten die folgende Fehlerantwort:

```
{
    "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.netfective.bluage.jac.server.services.rest.impl.JicsServiceImpl.createElement(JicsServiceI
```

Erstellen Sie eine TRANSAKTION

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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.netfective.bluage.jac.entities.JACTransaction`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TRANSAKTION erfolgreich im zugrunde liegenden JICS-Speicher erstellt.

Erstellen Sie ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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.netfective.bluage.jac.entities.JACProgram`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde das PROGRAMM erfolgreich im zugrunde liegenden JICS-Speicher erstellt.

Erstellen Sie eine DATEI

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 Objekts. `com.netfective.bluage.jac.entities.JACFile`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die DATEI erfolgreich im zugrunde liegenden JICS-Speicher erstellt.

Erstellen Sie eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 Objekts. `com.netfective.bluage.jac.entities.JACTDQueue`
- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die TDQUEUE erfolgreich im zugrunde liegenden JICS-Speicher erstellt.

Erstellen Sie ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 eine der 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, wurden die Elemente erfolgreich im zugrunde liegenden JICS-Speicher erstellt.

Aktualisieren Sie eine LISTE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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`

Die untergeordneten Elemente der LIST müssen nicht angegeben werden. Der LIST-Aktualisierungsmechanismus berücksichtigt die untergeordneten Elemente nicht.

- Gibt einen booleschen Wert zurück. Wenn der Wert 'true' ist, wurde die LISTE erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Wenn das LIST-Flag 'isActive' aktualisiert wird, wird es auf alle eigenen Elemente der LISTE übertragen, d. h. auf alle GRUPPEN, die sich im Besitz der LIST befinden, 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 eine der 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 erfolgreich im zugrunde liegenden JICS-Speicher 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 eine der 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 erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Aktualisieren Sie ein PROGRAMM

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Eine DATEI aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Aktualisieren Sie eine TDQUEUE

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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, TDQueue wurde er erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Aktualisieren Sie ein TSMODEL

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 erfolgreich im zugrunde liegenden JICS-Speicher aktualisiert.

Elemente aktualisieren

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 verärgern

- Unterstützte Methode: POST
- Erfordert Authentifizierung und eine der 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 eine der 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 eine der 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 Sie suchen. Dies ist die JSON-Serialisierung eines Objekts. `com.netfective.bluage.jac.entities.request.JicsCrudOperationRequest`
- Gibt eine JSON-Nutzlast zurück, die die Antwort darstellt. Dies ist die JSON-Serialisierung eines `com.netfective.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.netfective.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
- Pfad: `/api/services/security/servicelogin/userInfo`
- Argumente: Keine
- 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 zurück`com.netfective.bluage.jac.entities.SignOn`, serialisiert als JSON.

Löschen eines Benutzers

- Unterstützte Methode: POST
- Erfordert Authentifizierung und die Rolle ROLE_ADMIN.
- 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 für AWS Blu Age-Benutzer

Im folgenden Abschnitt erfahren Sie mehr über verschiedene Datenstrukturen für die AWS Blu Age Engine.

Themen

- Details zur Auftragsausführung, Nachrichtenstruktur
- 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

## Details zur Auftragsausführung, Nachrichtenstruktur

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 >= 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.netfective.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":"PVO",
     "jobName":"LOGGERJOB",
     "jobPriority":5,
```

```
    "jobQueue":"queue1",
    "scheduleMisfirePolicy": 4,
    "startTime": "2003/05/04 07:00:00.000 GMT-06:00",
    "endTime": "2003/05/04 07:00:07.000 GMT-06:00"
}
```

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 AS4 00 ist 5. Der gültige Bereich liegt zwischen 0 und 9, wobei 0 die höchste Priorität darstellt.

jobOnHold

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 Format MMddyy oder dd angegeben werden MMyyyy (die Größe der Eingabe bestimmt, welches Format verwendet wird)

Zeit

Kann mit Format HHmm oder angegeben werden HHmmss (die Größe der Eingabe bestimmt, welches Format verwendet wird)

ProgramParams

Wird als Map an das Programm übergeben.

scheduleMisfirePolicy

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 Endpunkts der the /schedule/list Jobwarteschlange.


```
[
  {
    "id": 1,
    "status": "ACTIVE",
    "jobNumber": 1,
    "userName": "PVO",
    "msg": {
      "messageQueueName": null,
      "scheduleDate": null,
      "scheduleTime": null,
      "startTime": "2024/03/07 21:12:00.000 UTC",
      "endTime": "2024/03/07 21:13:59.000 UTC",
      "programName": "LOGPGM",
      "programParams": {"cl_sbmjob_param_json": "[\"./output/schedule-job-log.txt\",
 \"Every 20 seconds!\"]"},
      "localDataAreaValue": "",
      "userName": "PVO",
      "jobName": "LOGGERJOB",
      "jobNumber": 1,
      "jobScheduleId": 1,
      "jobPriority": 5,
      "executionDate": null,
      "jobQueue": "queue1",
      "jobOnHold": false,
      "scheduleCron": "*/20 * * * * ?",
      "save": false,
      "scheduleMisfirePolicy": 4,
      "omitdates": null
    },
    "lastUpdatedAt": "2024-03-07T21:11:13.282+00:00",
    "lastUpdatedBy": ""
  }
]
```


# AWS Blu Age Runtime einrichten (nicht verwaltet)

In diesem Abschnitt werden die Schritte zur Einrichtung von AWS Blu Age Runtime (nicht verwaltet) auf Ihrer AWS Infrastruktur erläutert. Bevor Sie Ihre AWS Blu Age Runtime (nicht verwaltet) für

Anwendungen einrichten, sollten Sie sich mit den Voraussetzungen, Regionen und Bereichen sowie der CloudWatch Alarmeinstellung für die Konfiguration und Verwaltung Ihrer Laufzeitumgebung vertraut machen.

Themen


- AWS Voraussetzungen für Blu Age Runtime
- Einführung in AWS Blu Age Runtime
- Anforderungen an die Einrichtung der Infrastruktur für AWS Blu Age Runtime (nicht verwaltet)
- AWS Artefakte von Blu Age Runtime
- Stellen Sie AWS Blu Age Runtime auf Amazon bereit EC2
- Stellen Sie AWS Blu Age Runtime auf Containern auf Amazon ECS und Amazon EKS bereit
- Testen Sie die PlanetsDemo Anwendung


## AWS Voraussetzungen für Blu Age Runtime

AWS Blu Age Runtime (nicht verwaltet) ist in verschiedenen the section called “AWS Versionshinweise zu Blu Age” 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.

## Einführung in AWS Blu Age Runtime

Erstellen Sie zunächst einen AWS -SupportFall, um das Onboarding für den Zugriff auf AWS Blu Age Runtime anzufordern. Geben Sie in Ihrer Anfrage Ihre AWS-Konto ID, die AWS Region, die Sie verwenden möchten, und eine Computeroption sowie die AWS Blu Age Runtime-Version an. Wenn Sie sich nicht sicher sind, welche Version Sie benötigen, wenden Sie sich an Ihren AWS Blu Age-Liefermanager. Wenn Sie bereits über den Quellcode einer Anwendung verfügen, die mit den AWS Mainframe Modernization Refactoring-Tools generiert wurde, notieren Sie sich den `gapwalk.version` Tag-Wert in der `pom.xml` Datei in Ihrer modernisierten Codebasis.

> **Note**
>
> Die AWS Blu Age Runtime ist in zwei Hauptvarianten erhältlich: Alpha-Vorabversionen und Releases. Um herauszufinden, welche Version verwendet werden soll, wenden Sie the section called “AWS Blu-Age-Versionierung” sich an Ihren AWS Blu Age-Liefermanager oder wenden Sie sich an ihn.

### Regionen und Buckets für AWS Blu Age Runtime (nicht verwaltet)

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 AWS Blu Age Runtime (nicht verwaltet) zuzugreifen.

| AWS-Region | Bucket freigeben | Alpha-Vorabversions-Bucket |
|---|---|---|
| USA Ost (Ohio) | aws-bluage-runtime-artifacts-055777665268-US-Ost-2 | aws-bluage-runtime-artifacts-dev-055777665268-us-ost-2 |

| AWS-Region | Bucket freigeben | Alpha-Vorabversions-Bucket |
|---|---|---|
| USA Ost (Nord-Virginia) | aws-bluage-runtime-artifacts-139023371234-uns-Ost-1 | aws-bluage-runtime-artifacts-dev-139023371234-uns-Ost-1 |
| USA West (Nordkalifornien) | aws-bluage-runtime-artifacts-788454048782-us-west-1 | aws-bluage-runtime-artifacts-dev-788454048782-us-west-1 |
| USA West (Oregon) | aws-bluage-runtime-artifacts-836771190483-uns-West-2 | aws-bluage-runtime-artifacts-dev-836771190483-us-west-2 |
| Kanada (Zentral) | aws-bluage-runtime-artifacts-637423580979-ca-central-1 | aws-bluage-runtime-artifacts-dev-637423580979-ca-central-1 |
| Europa (Irland) | aws-bluage-runtime-artifacts-925278190477-eu-west-1 | aws-bluage-runtime-artifacts-dev-925278190477-eu-west-1 |
| Europa (London) | aws-bluage-runtime-artifacts-767397831990-eu-west-1 | aws-bluage-runtime-artifacts-dev-767397831990-eu-west-1 |
| Europa (Paris) | aws-bluage-runtime-artifacts-673009995881-eu-west-3 | aws-bluage-runtime-artifacts-dev-673009995881-eu-west-3 |
| Europa (Frankfurt) | aws-bluage-runtime-artifacts-485196800481-eu-central-1 | aws-bluage-runtime-artifacts-dev-485196800481-eu-central-1 |
| Europa (Stockholm) | aws-bluage-runtime-artifacts-654654484534-eu-nord-1 | aws-bluage-runtime-artifacts-dev-654654484534-eu-nord-1 |
| Europa (Milan) | aws-bluage-runtime-artifacts-654654328338-eu-Süd-1 | aws-bluage-runtime-artifacts-dev-654654328338-eu-süd-1 |
| Europa (Spain) | aws-bluage-runtime-artifacts-905417994954-eu-Süd-2 | aws-bluage-runtime-artifacts-dev-905417994954-eu-Süd-2 |
| Südamerika (São Paulo) | aws-bluage-runtime-artifacts-737536804457-sa-ost-1 | aws-bluage-runtime-artifacts-dev-737536804457-sa-ost-1 |

| AWS-Region | Bucket freigeben | Alpha-Vorabversions-Bucket |
|---|---|---|
| Asien-Pazifik (Tokio) | aws-bluage-runtime-artifacts-445578176276-ap-nordost-1 | aws-bluage-runtime-artifacts-dev-445578176276-ap-nordoast-1 |
| Asien-Pazifik (Seoul) | aws-bluage-runtime-artifacts-381492221498-ap-Nordost-2 | aws-bluage-runtime-artifacts-dev-381492221498-ap-Nordost-2 |
| Asien-Pazifik (Osaka) | aws-bluage-runtime-artifacts-905418229615-ap-Nordost-3 | aws-bluage-runtime-artifacts-dev-905418229615-ap-nordost-3 |
| Asien-Pazifik (Singapur) | aws-bluage-runtime-artifacts-767397774613-ap-Südosten-1 | aws-bluage-runtime-artifacts-dev-767397774613-ap-südosten-1 |
| Asien-Pazifik (Sydney) | aws-bluage-runtime-artifacts-726160321909-ap-Südosten-2 | aws-bluage-runtime-artifacts-dev-726160321909-ap-Südosten-2 |
| Asien-Pazifik (Mumbai) | aws-bluage-runtime-artifacts-905418353577-ap-süd-1 | aws-bluage-runtime-artifacts-dev-905418353577-ap-süd-1 |
| Afrika (Kapstadt) | aws-bluage-runtime-artifacts-992382777663-af-Süd-1 | aws-bluage-runtime-artifacts-dev-992382777663-af-süd-1 |
| Israel (Tel Aviv) | aws-bluage-runtime-artifacts-471112516508-il-central-1 | aws-bluage-runtime-artifacts-dev-471112516508-il-central-1 |

## 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 Ihren 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 Sie 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 EC2 Amazon-Instance zu aktualisieren.

`aws s3 cp s3://`*`bucket-name`*`/`*`version`*`/Framework/ `*`folder-of-your-choice`*` --recursive`

Wobei gilt:

folder-of-your-choice

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
```

> **Note**
>
> Der Zugriff auf Artefakte kann vorübergehend unterbrochen werden, und Versionen können aus Sicherheitsgründen entfernt werden. Wir empfehlen dringend, dass Sie die Artefakte, die Sie verwenden, in Ihrem eigenen Konto speichern. Die lokale Version sollte als Referenz in Ihren internen Architekturen verwendet werden.

## Anforderungen an die Einrichtung der Infrastruktur für AWS Blu Age Runtime (nicht verwaltet)

In diesem Thema wird die Mindestkonfiguration der Infrastruktur 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
- EC2 Amazon-Instanztypen für AWS Blu Age Runtime (bei Amazon EC2)
- AWS Blu Age Runtime auf Amazon ausführen EC2
- AWS Blu Age Runtime auf Amazon ECS auf Amazon ausführen EC2
- AWS Blu Age Runtime auf Amazon EKS auf Amazon ausführen EC2
- Ausführen von AWS Blu Age Runtime auf Amazon ECS, verwaltet von AWS Fargate

## Anforderungen an die Infrastruktur

### Eine Sicherheitsgruppe erstellen

Wenn Sie vorhaben, an EC2 Amazon-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 im 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 unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
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 Police 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 unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
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 EC2Rolle 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 in das Suchfeld den Namen der Richtlinie ein, die Sie zuvor erstellt haben.
7. Aktivieren Sie das Kontrollkästchen links neben Ihrer Richtlinie.

   > **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.

## EC2 Amazon-Instanztypen für AWS Blu Age Runtime (bei Amazon EC2)

Im Folgenden finden Sie eine Liste der EC2 Amazon-Instance-Typen, die Sie für AWS Blu Age Runtime (auf Amazon EC2) verwenden können, wenn Sie EC2 Amazon-Instances erstellen oder Amazon EKS-Worker-Knoten definieren.

Vergewissern Sie sich, dass die Instance, für die Sie sich interessieren, in der gewünschten Region verfügbar ist, die Sie bereitstellen möchten.

```
t3.small
t3.medium
t3.large
t3.xlarge
t3.2xlarge
t2.small
t2.medium
t2.large
```

```
t2.xlarge
t2.2xlarge
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 ausführen EC2

Gehen Sie wie folgt vor, um eine EC2 Amazon-Instance zu erstellen.

Eine EC2 Amazon-Instance erstellen

1. Öffnen Sie die EC2 Amazon-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 austhe section called “ EC2 Amazon-Instanztypen für AWS Blu Age Runtime (bei 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 die Option M2 BluagePrivateLink -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 EC2 Amazon-Instance

1. Wenn sich der Status der EC2 Amazon-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 (bei 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. Detaillierte Anweisungen finden Sie unter the section called “AWS Artefakte von Blu Age Runtime”.

## AWS Blu Age Runtime auf Amazon ECS auf Amazon ausführen EC2

1. Erstellen Sie einen Amazon ECS-Cluster mit  EC2 Amazon-Instances als zugrunde liegender 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 unter aufgeführten Instanztypen aus. the section called “ EC2 Amazon-Instanztypen für AWS Blu Age Runtime (bei Amazon EC2)”
4. Wählen Sie unter Netzwerkeinstellungen für EC2 Amazon-Instances die Sicherheitsgruppe aus, die Sie in den vorherigen Schritten erstellt haben.

## AWS Blu Age Runtime auf Amazon EKS auf Amazon ausführen EC2

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-Endpunkt 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 unter aufgeführten Instanztypen aus. the section called “ EC2 Amazon-Instanztypen für AWS Blu Age Runtime (bei Amazon EC2)”
5. Amazon EKS weist die Sicherheitsgruppe automatisch den gestarteten EC2 Amazon-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 Fargate im Amazon Elastic Container Service Developer Guide.

## AWS Artefakte von Blu Age Runtime

AWS Blu Age Runtime-Artefakte sind die Komponenten für die Bereitstellung und Ausführung modernisierter Anwendungen. In diesem Dokument werden die verschiedenen verfügbaren Artefakttypen, ihre Speicherorte und der Zugriff auf sie beschrieben.

### AWS Blu-Age Runtime-Artefakte (nicht verwaltet)

Zugriff auf und Speicherung von Artefakten

Die AWS Blu Age Runtime-Artefakte für nicht verwaltete Bereitstellungen werden in regionsspezifischen S3-Buckets gespeichert. Jede Version hat einen eigenen Ordner, der eine einfache Versionsverwaltung und einen einfachen Zugriff ermöglicht.

Es gibt zwei Arten von Buckets:

Eimer freigeben

Release-Buckets enthalten Verzeichnisse für die zuletzt bereitgestellten Versionen und folgen der Namenskonvention:. `aws-bluage-runtime-artifacts-<accountId>-<region>`

Buckets vor der Veröffentlichung

Pre-Release-Buckets enthalten Verzeichnisse für Alpha-Versionen, die den neuesten kurzlebigen inkrementellen Vorabversionen entsprechen, und folgen der Namenskonvention:. `convention: aws-bluage-runtime-artifacts-dev-<accountId>-<region>`

Der Zugriff auf die Produktions- und die Vorabversions-Buckets wird unabhängig voneinander gewährt. Weitere Informationen darüber, wie Sie Zugriff beantragen können, und weitere Informationen zur S3-Bucket-Organisation finden Sie unter. the section called “Einführung in AWS Blu Age Runtime ”

Inhalt der Artefakte

Sowohl im Release- als auch im Pre-Release-Bucket finden Sie:

aws-bluage-runtime-x.y.z.tar.gz

Dieses Archiv, wobei x.y.z für die Versionsnummer steht (`major.minor.patch`gemäß semantischer Versionierung, siehethe section called “AWS Blu-Age-Versionierung”), und enthält die wichtigsten AWS Blu Age-Runtime-Komponenten, die für die Ausführung von Blu Age-Anwendungen unerlässlich sind, darunter: AWS

- Gapwalk: Eine wichtige Komponente der AWS Blu Age Runtime, die entwickelt wurde, um die Lücke zwischen älteren Anwendungen und modernen Cloud-nativen Umgebungen zu schließen. Es dient als Kompatibilitätsebene, die es ermöglicht, dass von AWS Blu Age modernisierte Anwendungen effektiv auf modernen Plattformen ausgeführt werden können.
- bluage.bin: Die Kern-Binärdatei der AWS Blu Age Runtime. Diese Datei ist entscheidend für den Betrieb der Runtime.
- Alle notwendigen Bibliotheken und unterstützenden Dateien für den Betrieb von AWS Blu Age Runtime.

aws-bluage-webapps-x.y.z.tar.gz

Dieses Archiv, in dem x.y.z dem gleichen Versionsschema wie oben folgt, enthält Webanwendungen und Bibliotheken, die für die Verwaltung und Steuerung von Blu Age-Installationen erforderlich sind: AWS

- WAR-Datei (BAC) (Blusam Console), die zur Überwachung der Blusam-Datenbank verwendet wird.
- WAR-Datei (JAC Console), die zur Überwachung der JICS-Datenbank verwendet wird.
- Erforderliche unterstützende Bibliotheken.

Zusätzliche Dateien

- Prüfsummendateien, mit denen Sie die Integrität beider Blu Age-Archive gemäß der Benennungskonvention überprüfen können:
  - Für Runtime: `aws-bluage-runtime-x.y.z.tar.gz.checksumSHA256`
  - Für Webapps: `aws-bluage-webapps-x.y.z.tar.gz.checksumSHA256`
- In CVE-Berichtsdateien (nur für die Release-Versionen) werden die Inhalte dieser Version aufgeführt und die Benennungskonvention eingehalten: CVEs
  - Für Runtime: `Bluage-Runtime-x.y.z-CVEs.txt`
  - Für Webapps: `Bluage-Webapps-x.y.z-CVEs.txt`

Einzelheiten dazu, wie Sicherheitslücken behoben werden, finden Sie in der Versionsübersicht von AWS Mainframe Modernization Refactor with AWS Blu Age.

> **Note**
>
> Wir bemühen uns zwar, unsere Produkte ohne Produkte auf den Markt zu bringen, aber es CVEs kann sein CVEs, dass zu einem späteren Zeitpunkt neue hinzukommen. Die CVE-Berichtsdatei wird regelmäßig aktualisiert, um den neuesten Status widerzuspiegeln.

## Entwickler: AWS Blu Age Runtime Artifacts

Zugriff auf und Speicherung von Artefakten

Die Artefakte der AWS Blu Age Developer Runtime werden in speziellen S3-Buckets gespeichert. Diese Runtime umfasst sowohl Release- als auch Alpha-Vorabversionen. Der Zugriff auf diese Artefakte wird über Anfragen der AWS Blu Age-Toolbox verwaltet. Sobald Ihre Anfrage bearbeitet und genehmigt wurde, erhalten Sie Zugriff auf den entsprechenden Bereich aus dem in Ihrer Anfrage AWS-Konto angegebenen Bereich.

Runtime-Bucket für Entwickler

Der Haupt-Bucket für die Developer Runtime ist: `s3://toolbox-dev-runtime`

Ausführlichere Informationen zur Zugriffsanforderung und zum Verständnis der Bucket-Struktur finden Sie in der Dokumentation Dev and Special AWS Blu Age Runtimes.

Inhalt des Artefakts

Zu den Laufzeitartefakten für Entwickler gehören in der Regel:

gapwalk-x.y.z-dev.tar.gz

Dieses Archiv enthält die Entwicklungsversion der Gapwalk-Komponente, die ein wichtiger Bestandteil der Blu Age Runtime ist. AWS Es wurde entwickelt, um ältere Anwendungen mit modernen Cloud-nativen Umgebungen zu verbinden.

gapwalk-runtime-x.y.z-javadoc.zip

Diese Zip-Datei enthält die Dokumentation für die Gapwalk-Laufzeit. JavaDoc JavaDocbietet eine ausführliche API-Dokumentation, die besonders für Entwickler nützlich ist, die an der Integration oder Erweiterung der Gapwalk-Laufzeit arbeiten.

gapwalk-webapps-x.y.z-javadoc.zip

Ähnlich wie die Runtime JavaDoc enthält diese ZIP-Datei die JavaDoc Dokumentation speziell für die Gapwalk-Webanwendungen. Diese Dokumentation ist für Entwickler von entscheidender Bedeutung, die mit den webbasierten Komponenten des Gapwalk-Systems arbeiten oder diese anpassen.

## Stellen Sie AWS Blu Age Runtime auf Amazon bereit EC2

In den Themen in diesem Abschnitt erfahren Sie EC2, wie Sie AWS Blu Age Runtime (nicht verwaltet) bei Amazon einrichten, die Runtime-Version aktualisieren, Ihre Bereitstellung mithilfe von CloudWatch Amazon-Alarmen überwachen und lizenzierte Abhängigkeiten hinzufügen. Diese Anweisungen gelten sowohl für die Erstellung von EC2 Amazon-Instances als auch für die Verwendung von Amazon ECS auf Amazon EC2 oder Amazon EKS auf Amazon EC2.

Themen


- AWS Blu Age Runtime (nicht verwaltet) bei Amazon einrichten EC2
- Aktualisieren Sie die AWS Blu Age Runtime bei Amazon EC2
- AWS Blu Age Runtime (bei Amazon EC2) einrichten CloudWatch Amazon-Alarme
- Richten Sie lizenzierte Abhängigkeiten in AWS Blu Age Runtime auf Amazon ein EC2


### AWS Blu Age Runtime (nicht verwaltet) bei Amazon einrichten 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 Blu Age Runtime”Vollständig undthe section called “Einführung in AWS Blu Age Runtime ”.
- Erstellen Sie eine EC2 Amazon-Instance mit einem der unterstützten Instance-Typen. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EC2 Linux-Instances.
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur EC2 Amazon-Instance herstellen können, z. B. mithilfe von SSM.

> **Note**
>
> In diesem Handbuch wird davon ausgegangen, dass der Tomcat-Installationspfad `/m2-anywhere/tomcat-gapwalk/velocity` Stellen Sie sicher, dass Sie diesen Pfad verwenden, wenn Sie die folgenden Anweisungen befolgen, oder passen Sie die folgende Anweisung an den Pfad Ihrer Wahl an.

- Laden Sie AWS Blu Age Runtime herunter und entpacken Sie es (bei Amazon EC2). Kopieren Sie den Inhalt des Velocity-Verzeichnisses nach`/m2-anywhere/tomcat-gapwalk/velocity`. Stellen Sie sicher, dass Sie die `bluage.bin` Datei genau an dem Ort platzieren, der in der Umgebungsvariablen CATALINA_HOME angegeben ist, die in der Apache Tomcat-Dokumentation unter CATALINA_HOME und CATALINA_BASE beschrieben ist. Anweisungen zum Abrufen der AWS Blu Age Runtime-Artefakte, einschließlich Informationen zu Speicherung, Zugriff und Inhalt, finden Sie unter. the section called “AWS Artefakte von 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. Die AWS Blu Age Runtime führt das `PlanetsDemo-v1/jics/sql/initJics.sql` Skript beim ersten Start automatisch

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 EC2 Amazon-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 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. Navigieren Sie zum Verzeichnis `/m2-anywhere/tomcat-gapwalk/velocity /webapps/webapps`.
3. Kopieren Sie mit dem folgenden Befehl die im `PlanetsDemo-v1/webapps/` Ordner verfügbaren PlanetsDemo 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. `/m2-anywhere/tomcat-gapwalk/velocity /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.

```
/m2-anywhere/tomcat-gapwalk/velocity/startup.sh

tail -f /m2-anywhere/tomcat-gapwalk/velocity/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 unterthe section called “Testen Sie die PlanetsDemo Anwendung”.

## Aktualisieren Sie die AWS Blu Age Runtime bei Amazon EC2

In diesem Handbuch wird beschrieben, wie Sie die AWS Blu Age Runtime bei Amazon aktualisieren EC2.

### Themen

- Voraussetzungen
- Aktualisieren Sie die AWS Blu Age Runtime in der EC2 Amazon-Instanz
- 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 unterthe section called “ AWS Blu Age aktualisieren”.
- the section called “AWS Voraussetzungen für Blu Age Runtime”Vollständig undthe section called “Einführung in AWS Blu Age Runtime ”.

- Stellen Sie sicher, dass Sie über eine EC2 Amazon-Instance verfügen, die die neueste AWS Blu Age Runtime enthält. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EC2 Linux-Instances.
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur EC2 Amazon-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 Blu Age Runtime einrichten (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 EC2 Amazon-Instanz

Führen Sie die folgenden Schritte aus, um die AWS Blu Age Runtime zu aktualisieren.

1. Connect zu Ihrer EC2 Amazon-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. Detaillierte Anweisungen finden Sie unter the section called “AWS Blu Age Runtime (nicht verwaltet) bei Amazon einrichten 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 (bei Amazon EC2) einrichten CloudWatch Amazon-Alarme

Sie können CloudWatch den Empfang Ihres Anwendungsprotokolls einrichten und einen Alarm hinzufügen, der Sie vor möglichen Fehlern warnt. Auf diese Weise erhalten Sie besser sichtbare Benachrichtigungen, wenn Ihre bereitgestellten Anwendungen auf Ausnahmen stoßen. Die folgenden Abschnitte helfen Ihnen, die Konfiguration der CloudWatch Protokollierung und die Einrichtung von Alarmen zu verstehen und mehr darüber zu erfahren.

Bereitstellung der CloudWatch Protokollierung

Standardmäßig enthält die AWS Blu Age Runtime eine Protokolldatei mit dem Namen`logback-cloudwatch.yml`. Auf diese Datei wird in der `application-main.yml` Datei verwiesen, aber diese Referenz ist auskommentiert.

```
# logging:
#  config: classpath:logback-cloudwatch.xml
```

Beide Dateien befinden sich im Konfigurationsordner. Wenn Sie die obigen Zeilen auskommentieren, kann die Funktion aktiviert werden. CloudWatch Die Protokollierung kann konfiguriert werden, wie in den folgenden Abschnitten erläutert.

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.netfective.bluage.runtime.cloudwatchlogger.CloudWatchAppender">
        <logGroup>BluAgeRuntimeOnEC2-Logs</logGroup>
        <logStream>%date{yyyy-MM-dd,UTC}.%instanceId.%uuid</logStream>
        <layout>
            <pattern>%date{yyyy-MM-dd HH:mm:ss.SSS,UTC}  %level --- [%thread{15}]
 %logger{40} : %msg%n%xThrowable</pattern>
        </layout>
        <appender-ref ref="console" />
    </appender>

    <root level="INFO">
        <appender-ref ref="cloudwatch" />
    </root>
</configuration>
```

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 Instanz 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 Protokollgruppe `BluAgeRuntimeOnEC2-Logs` 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 Alarme konfigurieren. Insbesondere können Sie

proaktive Alarme 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 in der Verbindung mit dem AWS Blu Age-Kontrollsystem hervorgehoben. Anschließend können Sie einen Alarm definieren, der auf diese Metrik reagiert.

## Richten Sie lizenzierte Abhängigkeiten in AWS Blu Age Runtime auf Amazon ein EC2

In diesem Handbuch wird beschrieben, wie Sie zusätzliche lizenzierte Abhängigkeiten einrichten, die Sie mit AWS Blu Age Runtime auf Amazon verwenden können EC2.

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 Blu Age Runtime”Vollständig undthe section called “Einführung in AWS Blu Age Runtime ”.
- Stellen Sie sicher, dass Sie über eine EC2 Amazon-Instance verfügen, die die neueste AWS Blu Age Runtime (bei Amazon EC2) enthält. Weitere Informationen finden Sie unter Erste Schritte mit Amazon EC2 Linux-Instances.
- Stellen Sie sicher, dass Sie erfolgreich eine Verbindung zur EC2 Amazon-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 von AWS Blu Age Runtime (bei 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 (bei 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 (munity.jaspersoft)https://com. com/download-jaspersoft/community-Ausgabe). Wir haben die Funktionalität von AWS Blu Age Runtime (bei 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 EC2 Amazon-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-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.

## Stellen Sie AWS Blu Age Runtime auf Containern auf Amazon ECS und Amazon EKS bereit

In den Themen in diesem Abschnitt erfahren Sie, wie Sie AWS Blu Age Runtime auf Containern einrichten, um sie auf Amazon ECS (verwaltet von Amazon EC2 oder AWS Fargate) und Amazon EKS, verwaltet von Amazon EC2, zu aktualisieren, die Runtime-Version zu aktualisieren,

Ihre Bereitstellung mithilfe von CloudWatch Amazon-Alarmen zu überwachen und lizenzierte Abhängigkeiten hinzuzufügen.

> **Note**
>
> Dies ist nicht kompatibel mit Amazon EKS, verwaltet von AWS Fargate.

Themen


- Richten Sie AWS Blu Age Runtime auf dem Container ein
- Aktualisieren Sie die AWS Blu Age Runtime auf dem Container
- CloudWatch Amazon-Alarme für AWS Blu Age Runtime auf dem Container einrichten
- Richten Sie lizenzierte Abhängigkeiten in AWS Blu Age Runtime auf dem Container ein


## Richten Sie AWS Blu Age Runtime auf dem Container ein

In diesem Thema wird erklärt, wie Sie die PlanetsDemo Beispielanwendung mithilfe von AWS Blu Age Runtime auf einem Docker-Container einrichten und bereitstellen.

AWS Blu Age Runtime on Container ist für Amazon ECS, verwaltet von Amazon EC2, Amazon ECS verwaltet von und Amazon EKS AWS Fargate, verwaltet von Amazon, verfügbar EC2. Es ist nicht kompatibel mit Amazon EKS, verwaltet von AWS Fargate.

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 Blu Age Runtime”Vollständig undthe section called “Einführung in AWS Blu Age Runtime ”.
- Laden Sie die AWS Blu Age Runtime-Binärdateien herunter. Detaillierte Anweisungen finden Sie unter the section called “Einführung in AWS 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-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 sind die Inhalte von `install-gapwalk.sh`.

```


# Vars
TEMP_DIR=/bluage-on-fargate/tomcat.gapwalk/temp
```

```
# Install
echo "Installing Gapwalk and Tomcat"
sudo rm -rf /bluage-on-fargate
mkdir -p ${TEMP_DIR}
# Copy Blu Age runtime and tomcat archives to temporary extraction dir
sudo cp /usr/local/velocity/installation/gapwalk/gapwalk.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-on-fargate/tomcat.gapwalk/velocity/webapps/*
# Extract Blu Age runtime at velocity dir
tar -xvf ${TEMP_DIR}/gapwalk.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-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. Erstellen Sie dann je nach Situation entweder einen Amazon EKS-Pod oder eine Amazon ECS-Aufgabendefinition mit Ihrem Amazon ECR-Image und stellen Sie es in Ihrem Cluster bereit. Beispiele für deren Erstellung finden Sie unter Erstellen einer Aufgabendefinition mithilfe der Konsole im Amazon Elastic Container Service (Amazon ECS) Developer Guide und Deploy a sample application im Amazon EKS-Benutzerhandbuch.
6. Insbesondere für Amazon ECS Managed by AWS Fargate Case sollten Sie bei der Erstellung der Aufgabendefinition die IAM-Rolle verwenden, die Sie im Rahmen der anfänglichen Einrichtung der Infrastruktur erstellt haben. Erweitern Sie dann bei der Erstellung des Dienstes den Abschnitt Netzwerk und konfigurieren Sie die VPC, die Subnetze und die Sicherheitsgruppe, die Sie im Rahmen der ersten Einrichtung der Infrastruktur erstellt haben. Weitere Informationen finden Sie unter Anforderungen für die Einrichtung der Infrastruktur für AWS Blu Age Runtime (nicht verwaltet).

Testen Sie die bereitgestellte Anwendung

Ein Beispiel für das Testen der PlanetsDemo Anwendung finden Sie unterthe section called “Testen Sie die PlanetsDemo Anwendung”.

## Aktualisieren Sie die AWS Blu Age Runtime auf dem Container

In diesem Handbuch wird beschrieben, wie Sie die AWS Blu Age Runtime auf einem Container aktualisieren. Dazu müssen Sie zunächst einige Voraussetzungen erfüllen und dann mit dem Docker-Image arbeiten, um die AWS Blu Age Runtime zu aktualisieren.

Themen

- Voraussetzungen
- Aktualisieren Sie die AWS Blu Age Runtime

Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- the section called “AWS Voraussetzungen für Blu Age Runtime”Vollständig undthe section called “Einführung in AWS 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 “Einführung in AWS 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. Detaillierte Anweisungen finden Sie unter the section called “Richten Sie AWS Blu Age Runtime auf dem Container ein”.
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.

## CloudWatch Amazon-Alarme für AWS Blu Age Runtime auf dem Container einrichten

Sie können festlegen CloudWatch , dass Benachrichtigungen besser sichtbar sind, wenn bei Ihren bereitgestellten Anwendungen Ausnahmen auftreten. Auf diese Weise können Sie Ihr Anwendungsprotokoll überwachen CloudWatch, an das umgeleitet wird, und einen Alarm hinzufügen, 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 bei der Clustererstellung 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 den Protokollen eine Metrik hinzu, die sich auf die Zeichenfolge „Fehler C“ bezieht. Anschließend können Sie einen Alarm definieren, der auf diese Metrik reagiert.

# Richten Sie lizenzierte Abhängigkeiten in AWS Blu Age Runtime auf dem Container ein

In diesem Thema wird beschrieben, wie Sie zusätzliche lizenzierte Abhängigkeiten einrichten, die Sie mit AWS Blu Age Runtime auf einem Container verwenden können.

Themen


- Voraussetzungen
- Übersicht


## Voraussetzungen

Bevor Sie beginnen, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- the section called “AWS Voraussetzungen für Blu Age Runtime”Vollständig undthe section called “Einführung in AWS 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 (munity.jaspersoft)https://com. com/download-jaspersoft/community-Ausgabe). 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-Datenbank auf Oracle gehostet wird, stellen Sie den Oracle-Datenbanktreiber unter bereit. *your-tomcat-path*`/extra`
3. Kopieren Sie diese Abhängigkeiten in 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- oder Amazon EKS-Service und starten Sie ihn neu.
6. Überprüfen Sie die Protokolle.

## Testen Sie die PlanetsDemo Anwendung

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
```

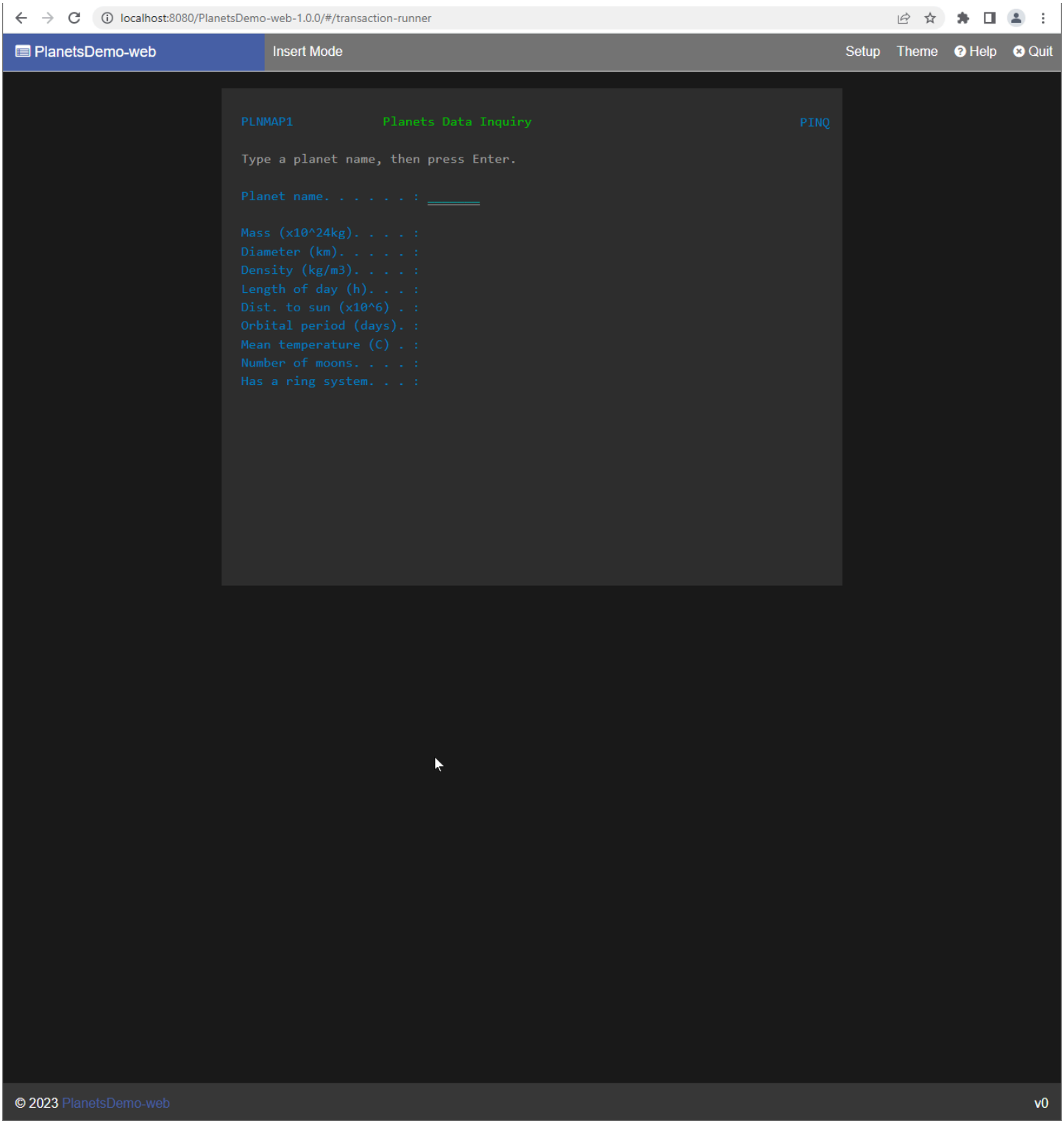
Notieren Sie sich den Namen der Web-Binärdatei (PlanetsDemo-web-1.0.0, falls unverändert). Verwenden Sie eine URL des PlanetsDemo folgenden Formats, um auf die Anwendung zuzugreifen.

```
https://load-balancer-DNS-name:listener-port/web-binary-name
```

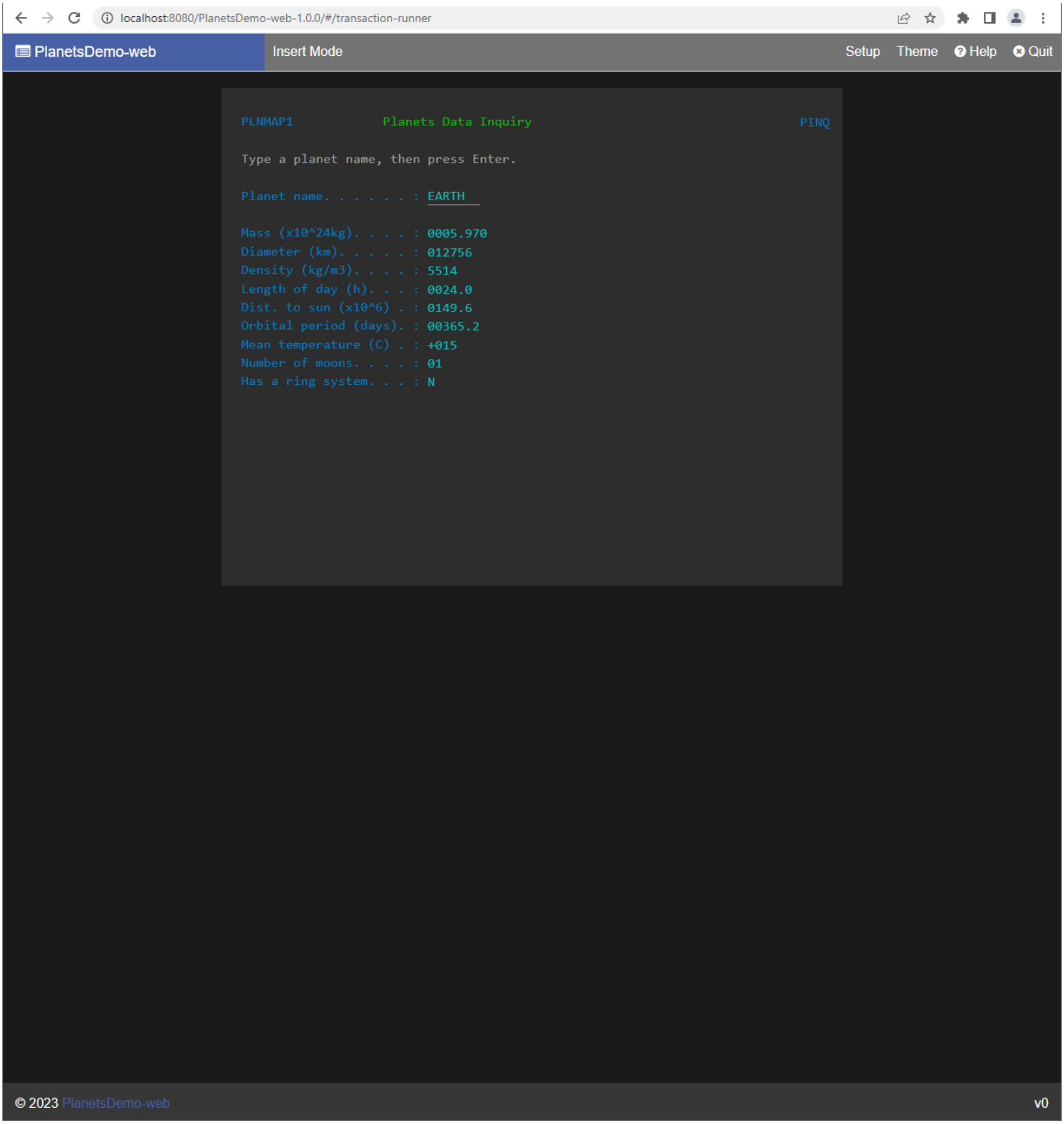
Nach dem Start der PlanetsDemo Anwendung wird die Startseite angezeigt.




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.




AWS Blu Age Runtime ist in den folgenden Regionen verfügbar: USA Ost (Ohio), USA Ost (Nord-Virginia), USA West (Nordkalifornien), USA West (Oregon), Kanada (Zentral), Region Europa (Irland), Region Europa (London), Region Europa (Paris), Europa (Frankfurt), Region Europa (Stockholm), Region Europa (Mailand), Region Europa (Spanien), Südamerika (São Paulo), Asien-Pazifik (Tokio),

Asien-Pazifik (Seoul), Asien-Pazifik (Osaka), Asien-Pazifik (Singapur), Asien-Pazifik (Sydney), Asien-Pazifik (Mumbai), Afrika (Kapstadt) und Israel (Tel Aviv).

# 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 Repositorys 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


- Voraussetzungen
- 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


## Voraussetzungen

Gehen Sie für Erstbenutzer wie folgt vor:

1. Navigieren Sie zu https://console.aws.amazon.com/appstream2/Hause zur AppStream 2.0-Konsole.
2. Wählen Sie Get Started.
3. Klicken Sie auf Skip.

> **Important**
>
> Amazon AppStream 2.0 verwendet IAM-Rollen zur Verwaltung Ihrer AppStream 2.0-Ressourcen und erstellt AWS diese Rollen, wenn Sie dies tun.

Laden Sie anschließend 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 EC2 Amazon-Instance, um die Upload- und Download-Leistung zu verbessern. Weitere Informationen zum Starten und Herstellen einer Verbindung zu einer EC2 Amazon-Instance finden Sie unter Erste Schritte mit Amazon EC2.

### 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. Weitere Informationen zu Buckets finden Sie unter Buckets erstellen.

### 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. Weitere Informationen zum Anhängen einer Richtlinie an einen S3-Bucket finden Sie unter Hinzufügen einer Bucket-Richtlinie.

Stellen Sie sicher, dass Sie es `amzn-s3-demo-bucket` 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:::amzn-s3-demo-bucket/*"
    }]
}
```

### 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.

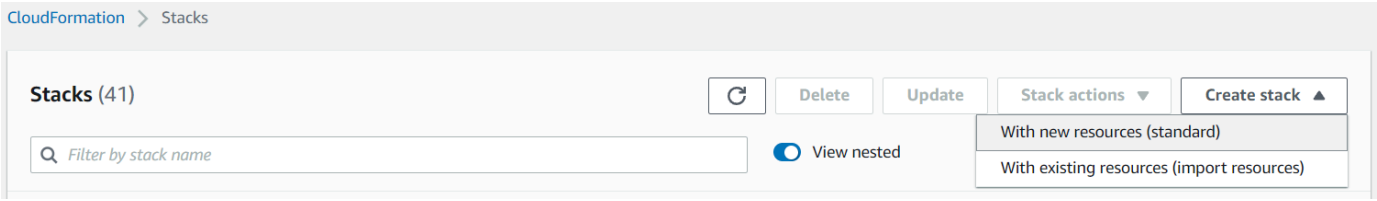
- cfn-m2-.yaml appstream-elastic-fleet-linux

- 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

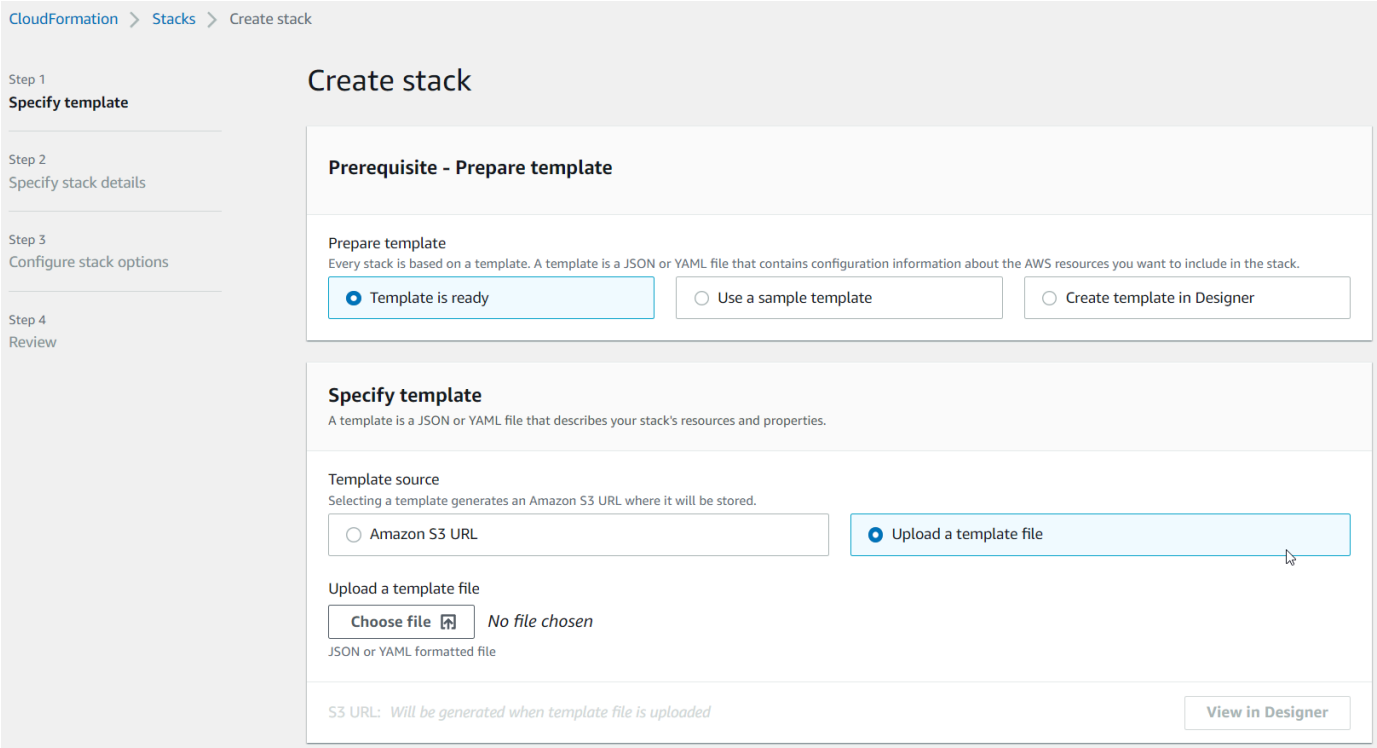
## Schritt 5: Erstellen Sie die Flotte mit AWS CloudFormation

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:




3. Wählen Sie unter Stapel erstellen die Option Vorhandene Vorlage auswählen und Vorlagendatei hochladen aus:




4. Wählen Sie Datei auswählen und navigieren Sie zur Datei`cfn-m2-appstream-elastic-fleet-linux.yaml`. Wählen Sie Weiter aus.
5. Geben Sie im Feld Stapeldetails angeben 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.

6. Wählen Sie Weiter aus.
7. Navigieren Sie auf der Seite nach unten und wählen Sie Ich bestätige, dass AWS CloudFormation möglicherweise IAM-Ressourcen mit benutzerdefinierten Namen erstellt werden. .
8. Wählen Sie Weiter aus.
9. Überprüfen Sie die Details und wählen Sie Senden.
10. Nachdem Sie die Flotte erstellt haben, erstellen Sie CloudFormation Stapel mit allen 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.

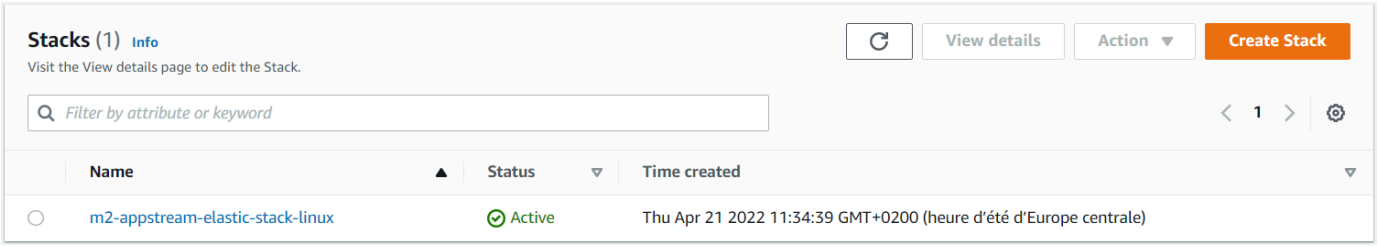
> **Note**
>
> Die heruntergeladenen Vorlagen erwarten, dass sie Assets in einem S3-Bucket mit einer Ordnerstruktur namens finden`appstream/bluage/developer-ide/`. Der Bucket muss sich in derselben Flotte AWS-Region befinden, die Sie erstellt haben.

> **Important**
>
> Führen Sie alle in Schritt 4 heruntergeladenen CloudFormation Skripts aus, um Ihre Anwendung korrekt einzurichten.

## 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:




2. Wählen Sie auf der Seite mit den Stack-Details den Stack und dann Associate fleet aus.
3. Wählen Sie in der Eingabeaufforderung die Flotte aus, die Sie zuvor erstellt und gestartet haben.
4. Wählen Sie Associate aus.
5. Wählen Sie den zugehörigen Stack aus und wählen Sie im Menü Aktionen die Option Streaming-URL erstellen, geben Sie 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 nativen 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.
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.

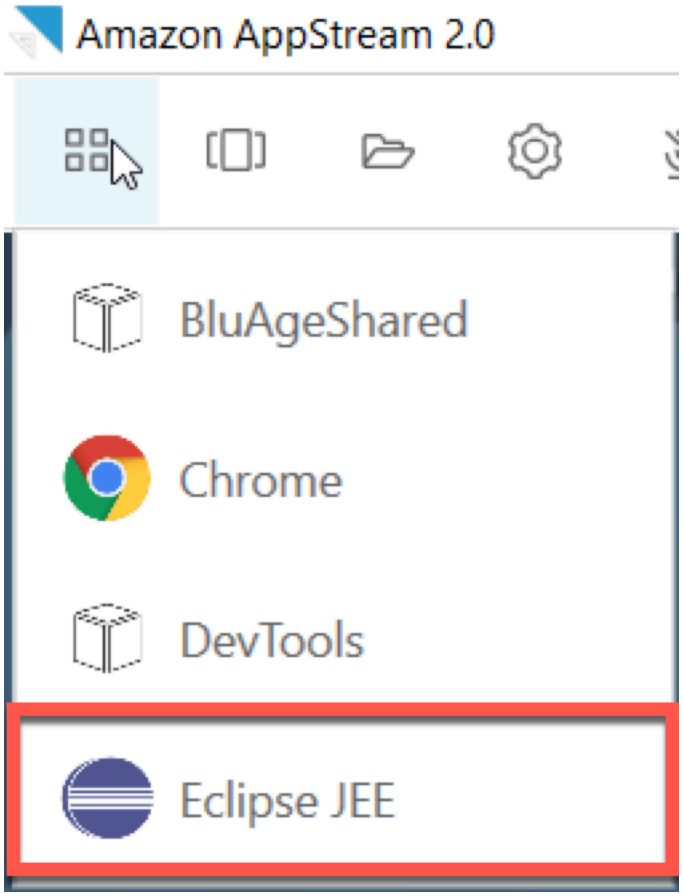
1. Wählen Sie in der Menüleiste Anwendungen und dann Terminal aus.
2. Geben Sie den folgenden Befehl ein:

```
~/_install-velocity-runtime.sh
```

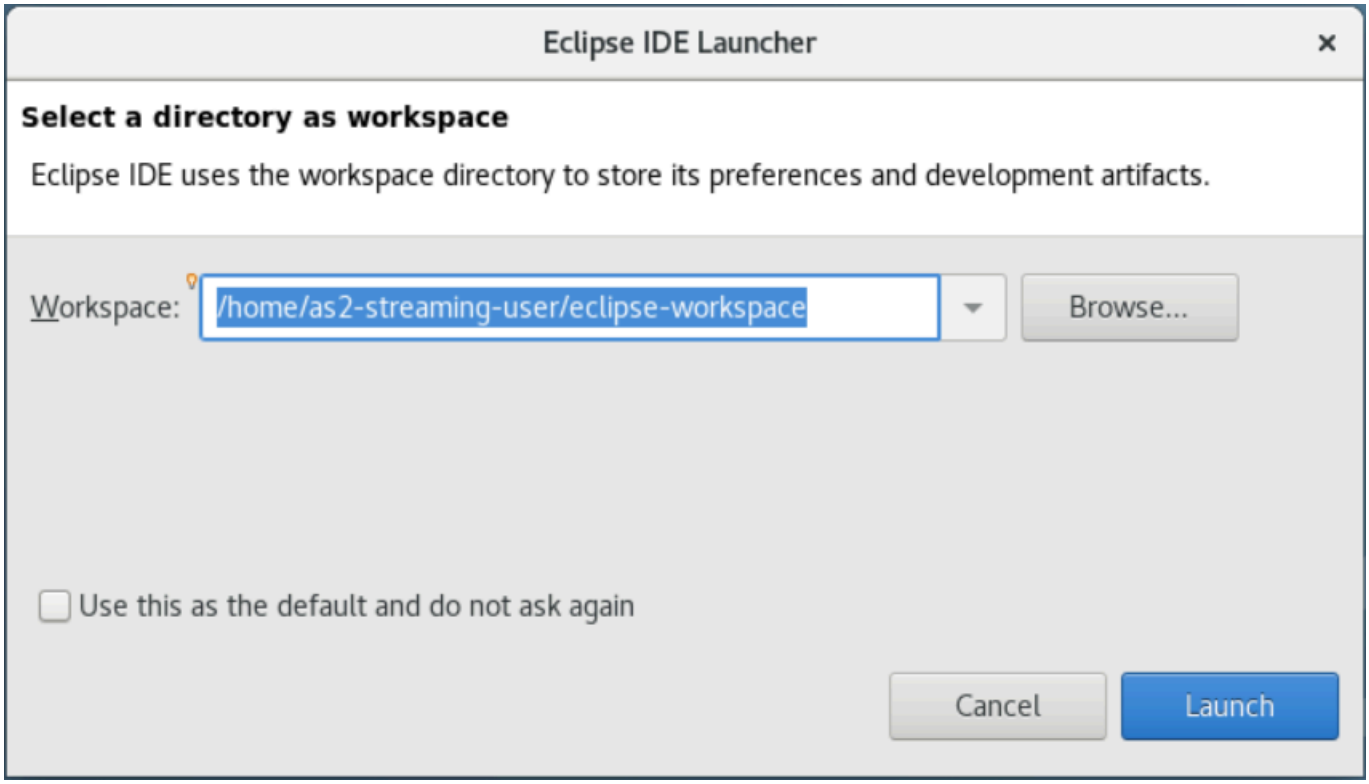
## Schritt 4: Starten Sie die Eclipse-IDE

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:

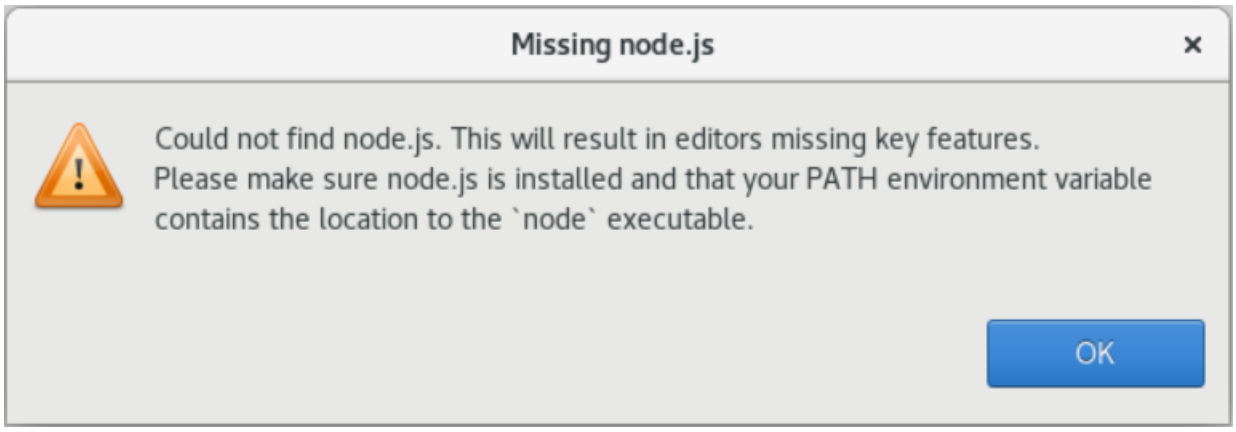
```
cat ~/runtime-version.txt
```

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 herunter`node.js`, um den Angular-Teil (*-Web) des Projekts zu erstellen:

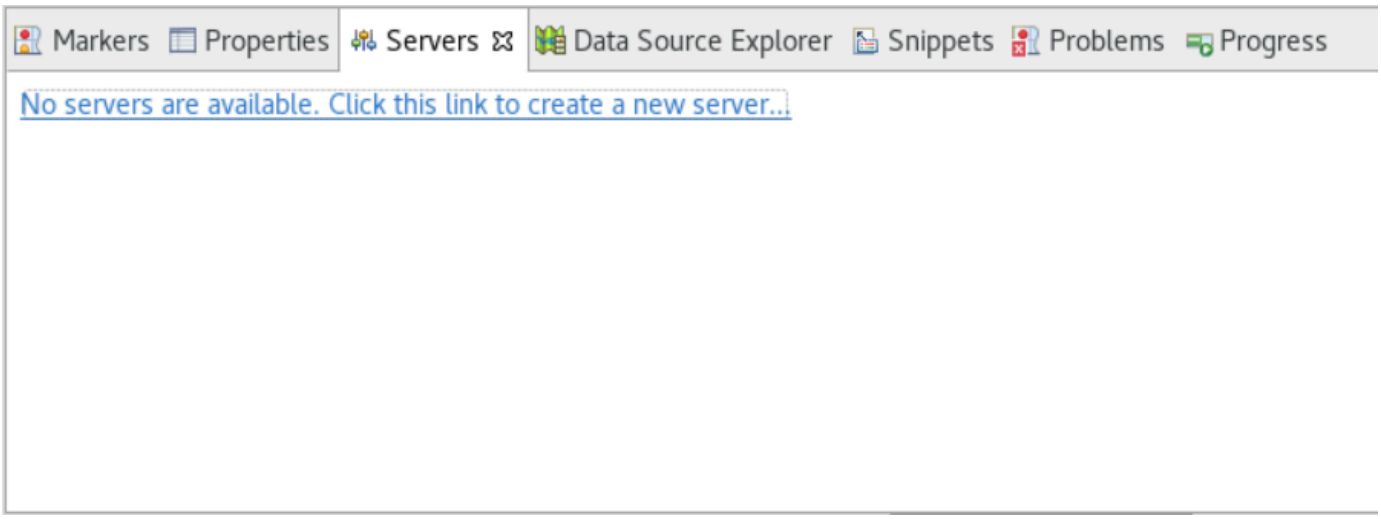
   
   

   Warte bis zum Ende des Builds. Sie können den Build in der Fortschrittsansicht verfolgen.
7. Wählen Sie in Eclipse das Projekt aus und wählen Sie Ausführen als. Wählen Sie dann Maven install. Nachdem die Maven-Installation erfolgreich war, erstellt sie die `war` Datei unter. `PlanetsDemoPom/PlanetsDemo-web/target/PlanetsDemo-web-1.0.0.war`

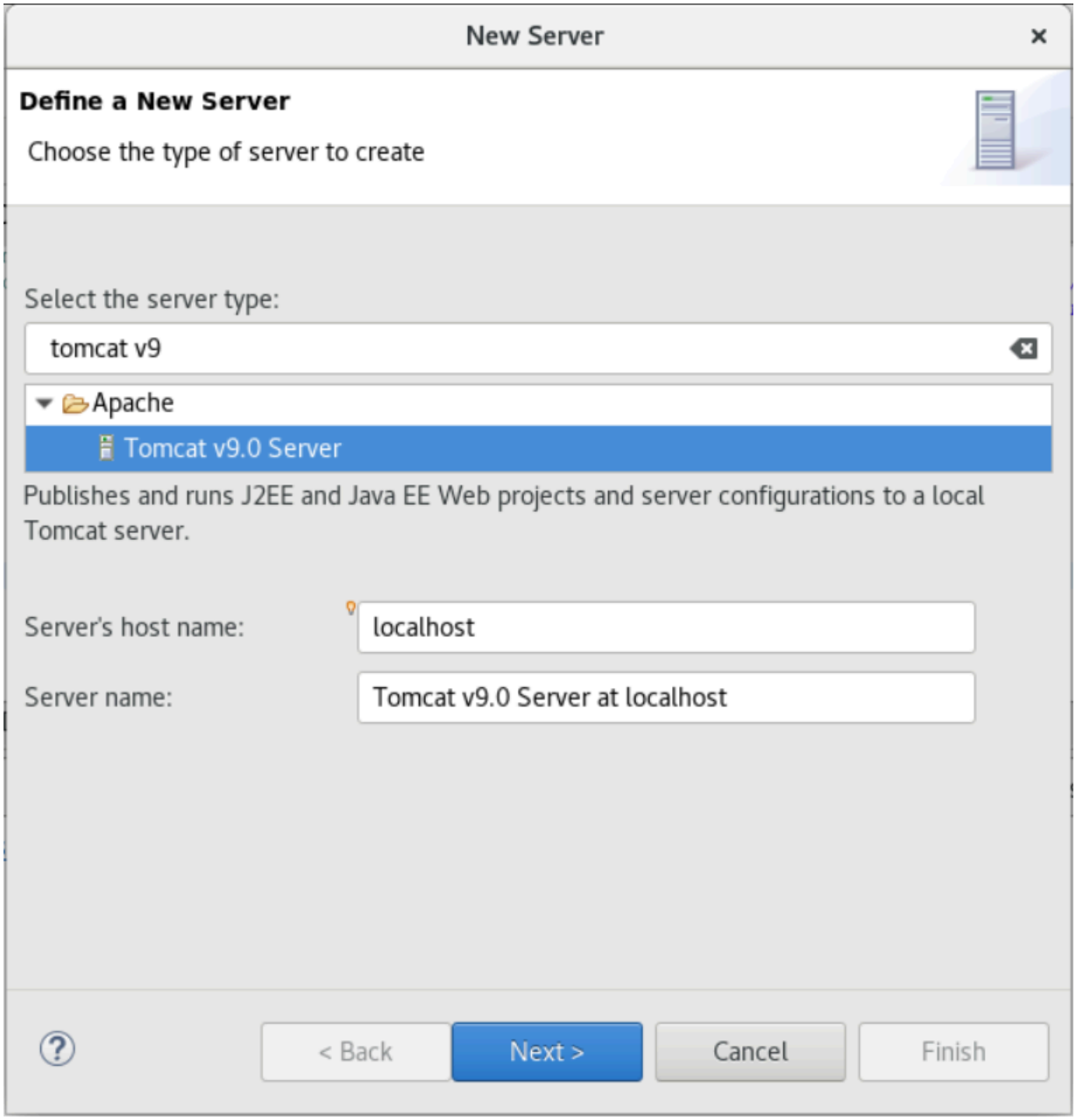
## Schritt 6: Konfigurieren Sie einen Tomcat-Server

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:

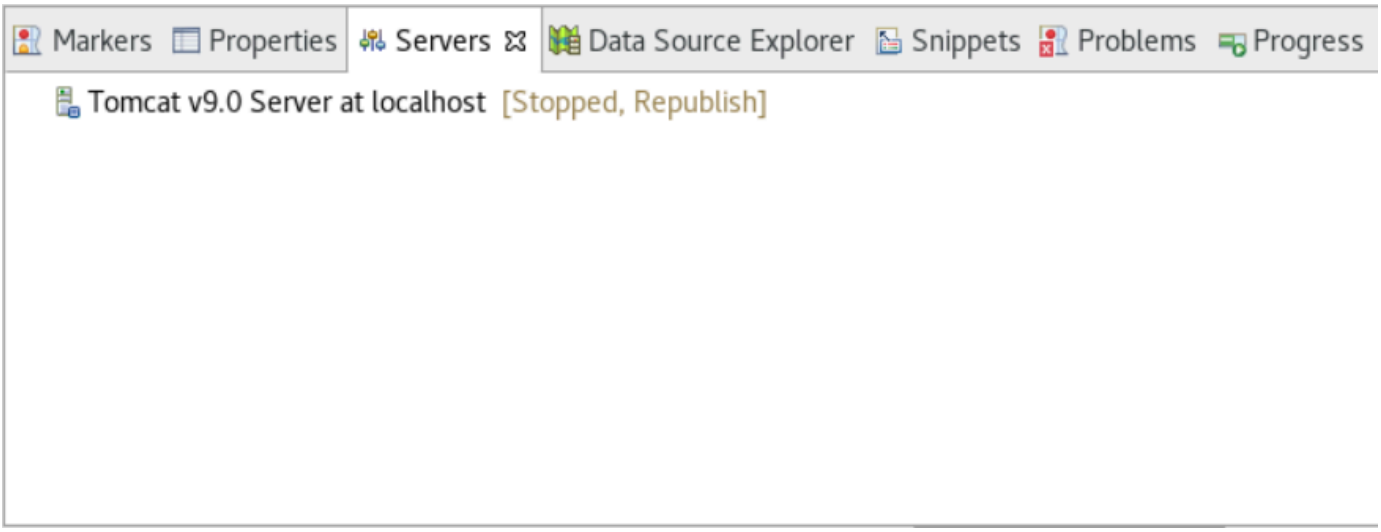



2. Wählen Sie Es sind keine Server verfügbar. Klicken Sie auf diesen Link, um einen neuen Server zu erstellen... . Der Assistent für neue Server wird angezeigt. Geben Sie im Feld Servertyp auswählen des Assistenten tomcat v9 ein und wählen Sie Tomcat v9.0 Server aus. Wählen Sie anschließend Weiter.




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“.

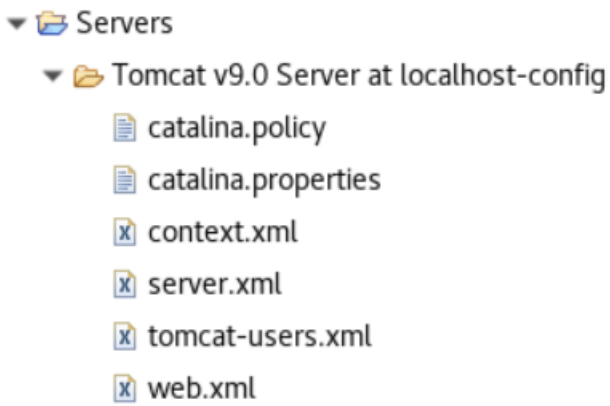
   Ein Serverprojekt wird im Workspace erstellt, und ein Tomcat v9.0-Server ist jetzt in der Serveransicht verfügbar. Hier wird die kompilierte Anwendung bereitgestellt und gestartet:



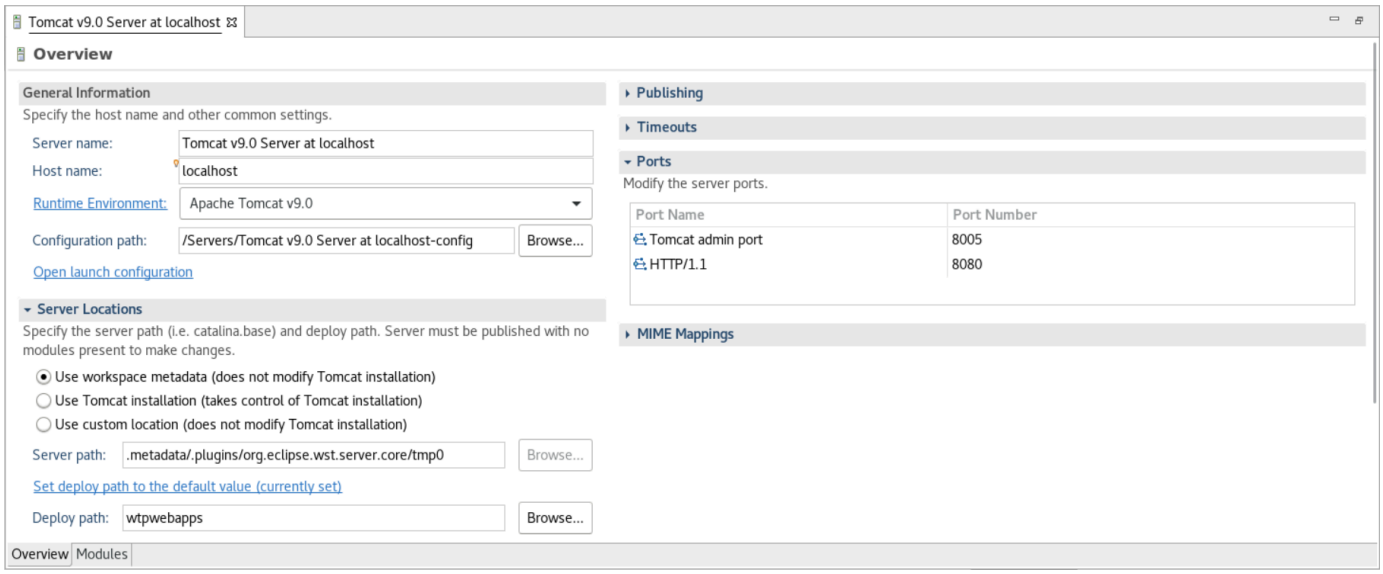

## Schritt 7: Auf Tomcat bereitstellen

In diesem Schritt stellen Sie die Planets-Demo-Anwendung auf dem Tomcat-Server bereit, damit Sie die Anwendung ausführen können.

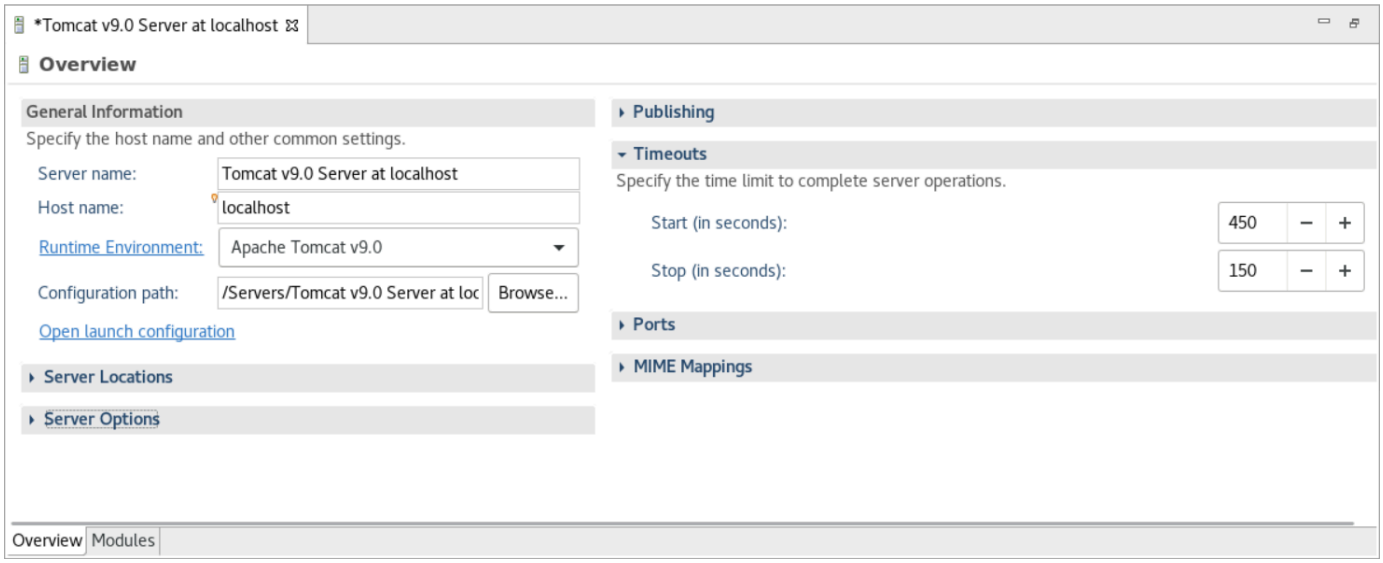
1. Wählen Sie die `PlanetsDemo-web` Datei aus und wählen Sie „Ausführen als“ > „Maven-Installation“. Wählen Sie `PlanetsDemo-web` erneut und wählen Sie Refresh, um sicherzustellen, dass das npm-kompilierte Frontend ordnungsgemäß in die Datei .war kompiliert wurde und von Eclipse erkannt wird.
2. Laden 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:



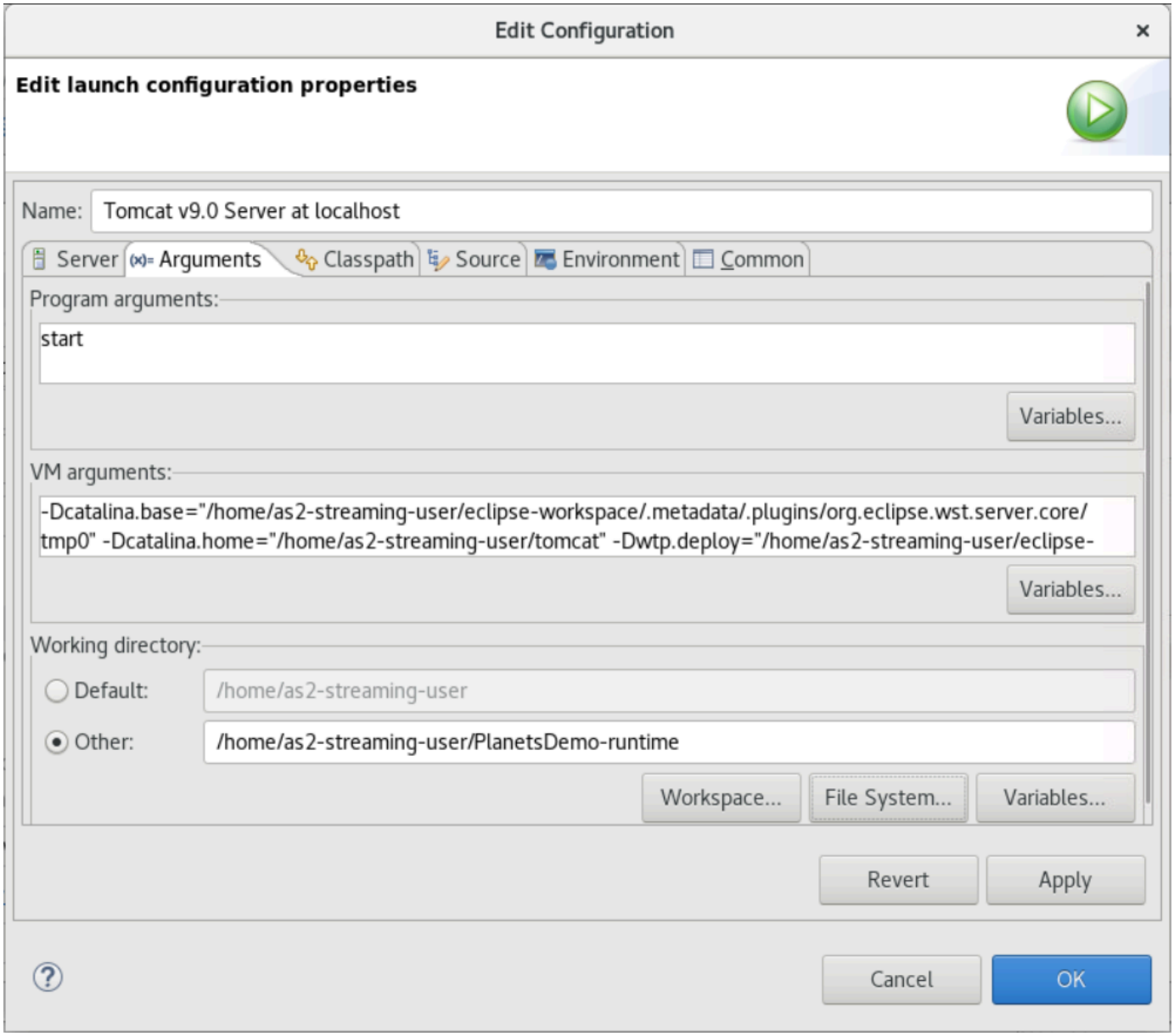
4. Öffnen Sie den `tomcat v9.0` Servereintrag in der Serveransicht. Der Servereigenschaften-Editor wird angezeigt:




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.

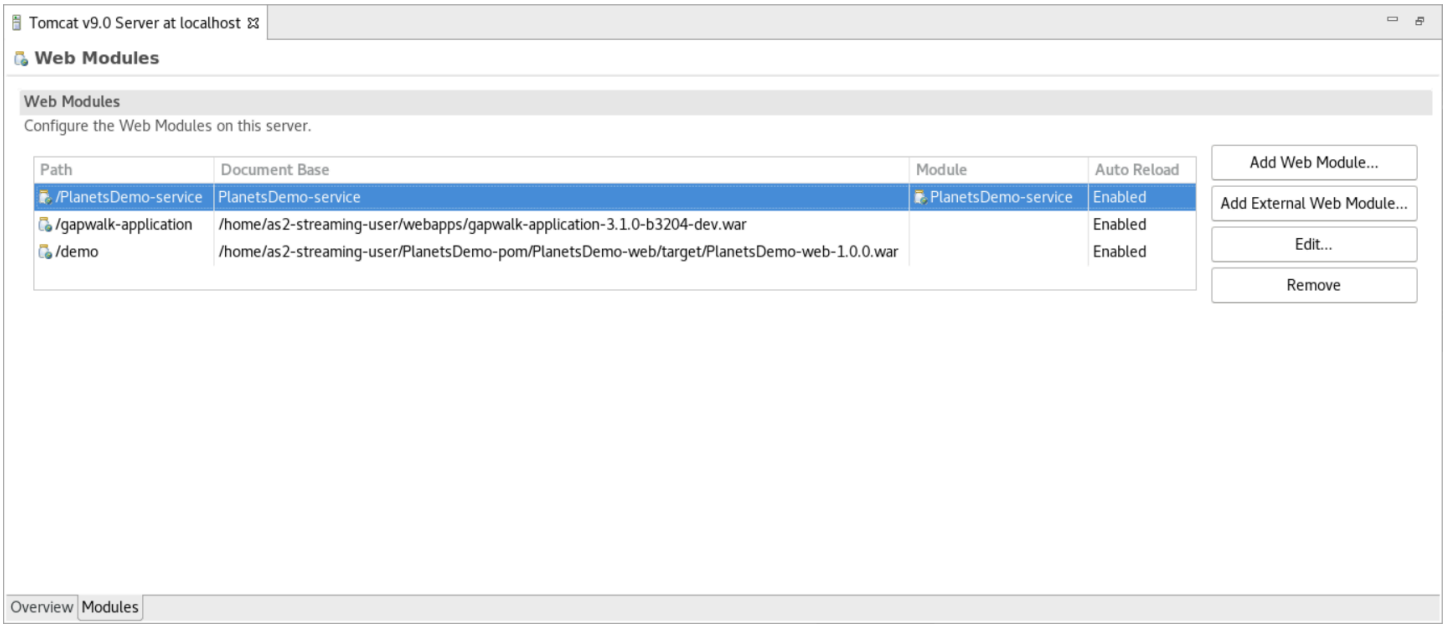



7. Wählen Sie im Servereigenschaften-Editor die Registerkarte Module und nehmen Sie die folgenden Änderungen vor:

   - Wählen Sie „Web-Modul hinzufügen“ und „Hinzufügen`PlanetsDemo-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

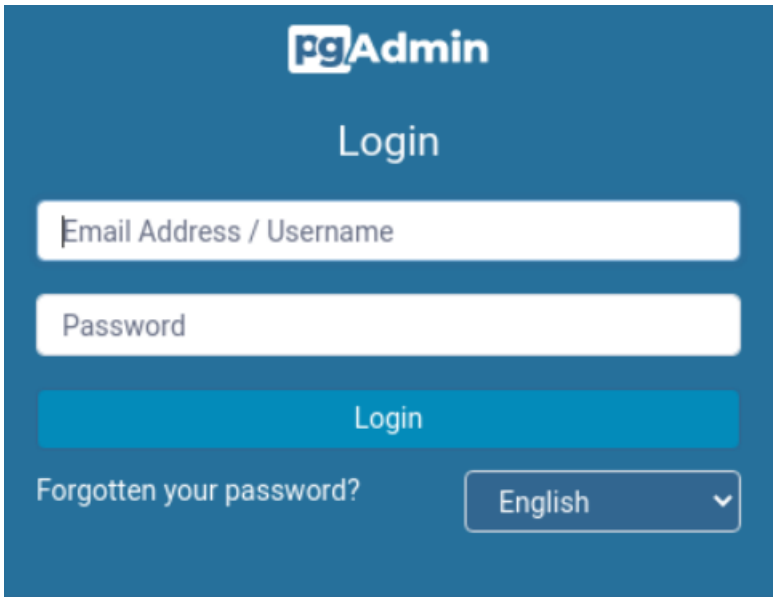
In diesem Schritt stellen Sie eine Verbindung zu der Datenbank her, in [Schritt 1: Erstellen einer Datenbank](#) der Sie erstellt haben.

1. Geben Sie in der AppStream 2.0-Instance den folgenden Befehl in einem Terminal ein, um ihn zu startenpgAdmin:

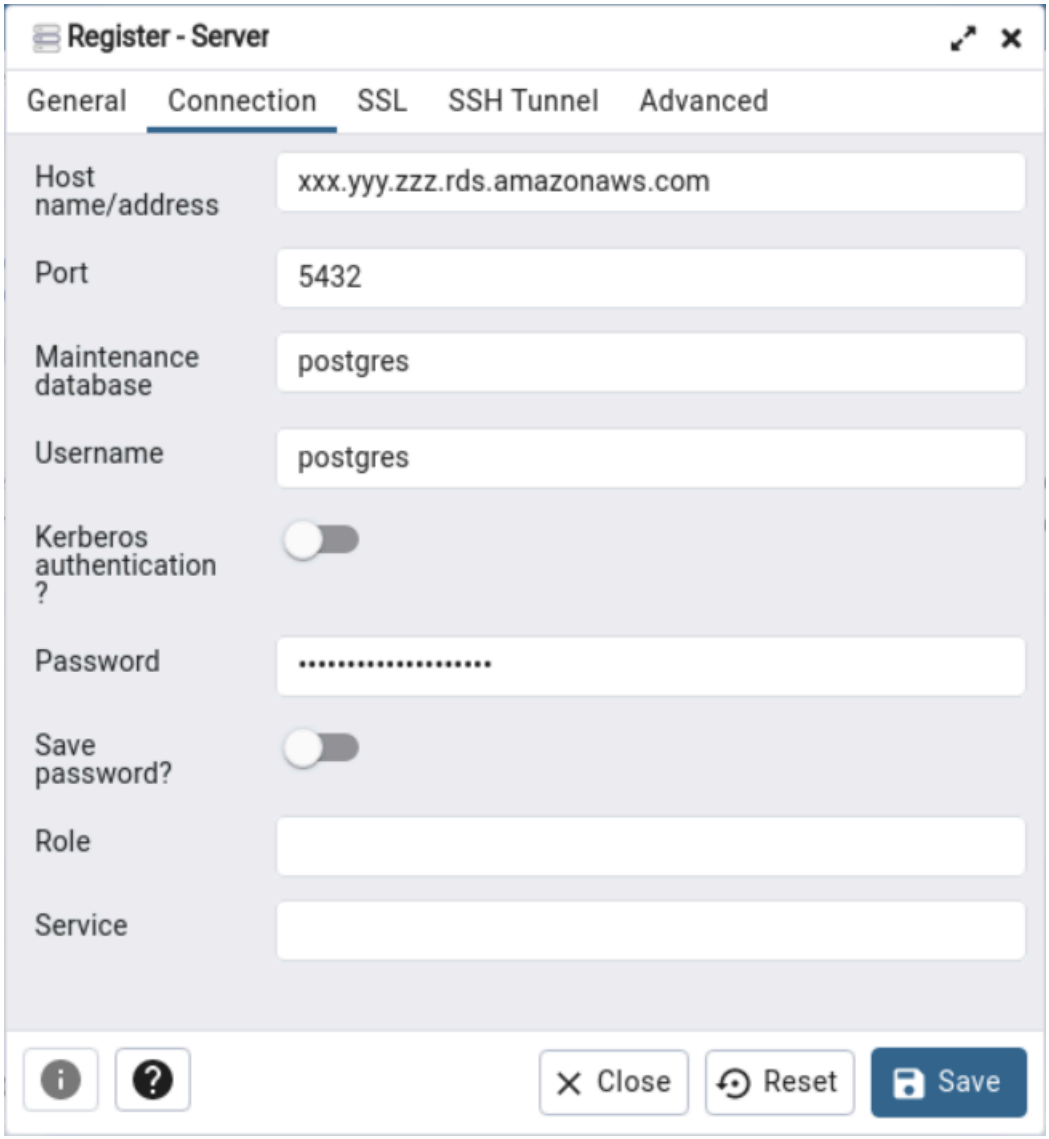
   ```
   ./pgadmin-start.sh
   ```

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.

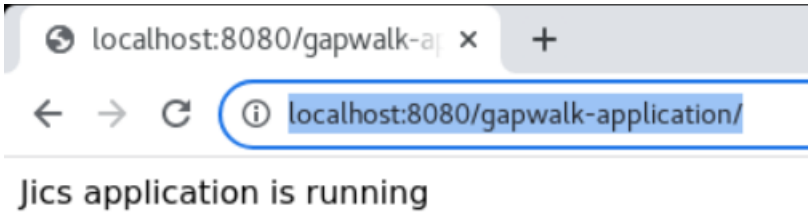



4. Wenn Sie eine Verbindung zum Datenbankserver herstellen, verwenden Sie Objekt > Erstellen > Datenbank und erstellen Sie eine neue Datenbank mit dem Namen jics.
5. Bearbeiten Sie die Datenbankverbindungsinformationen, die die Demo-App verwendet hat. Diese Informationen sind in definiert`PlanetsDemo-runtime/config/application-main.yml`. Suchen Sie nach dem `jicsDs` Eintrag. Um die Werte für abzurufen `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.

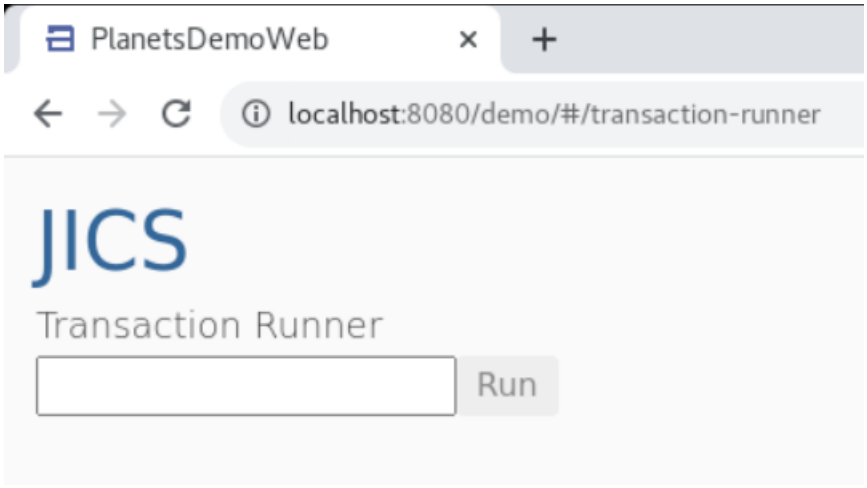
## Schritt 9: Starten und testen Sie die Anwendung

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.

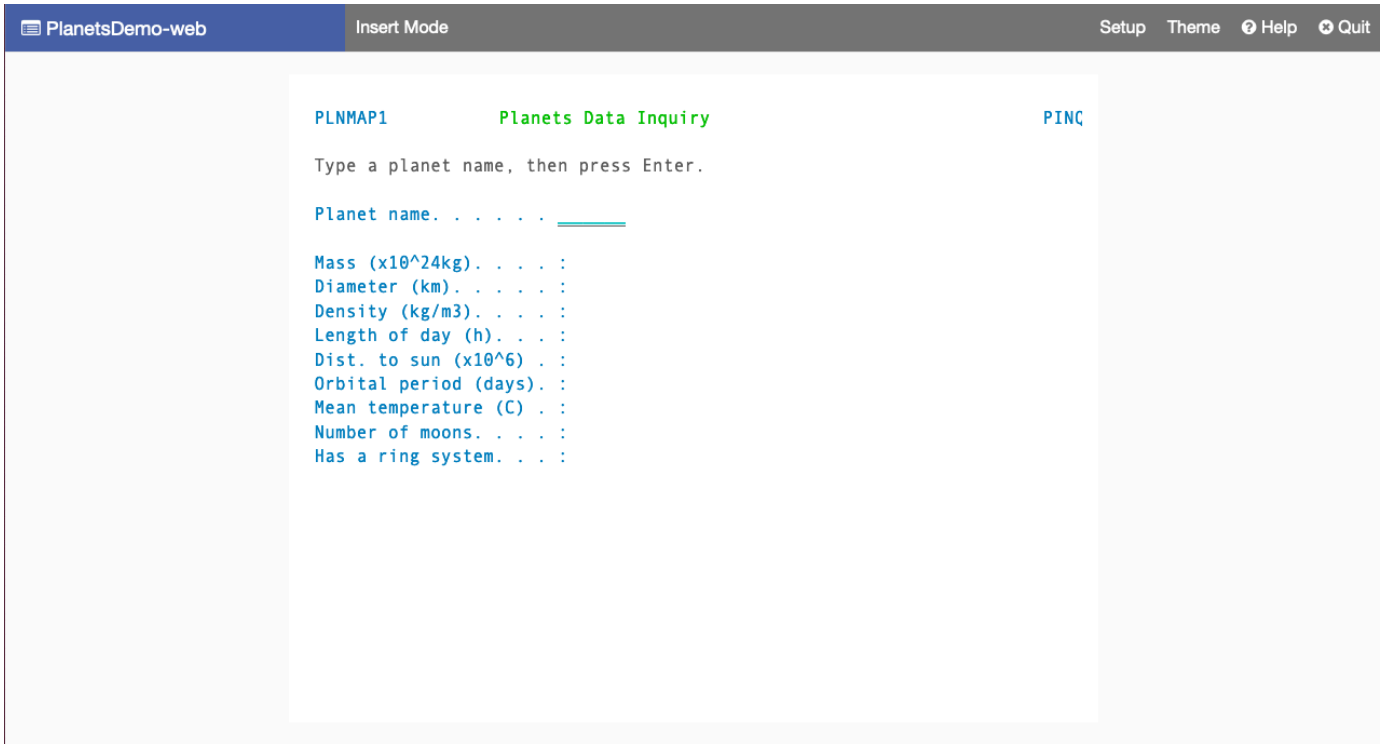



3. Greifen Sie über Google Chrome unter http://localhost:8080/demo auf das bereitgestellte Anwendungs-Frontend zu. Die folgende Transaction Launcher-Seite sollte angezeigt werden.

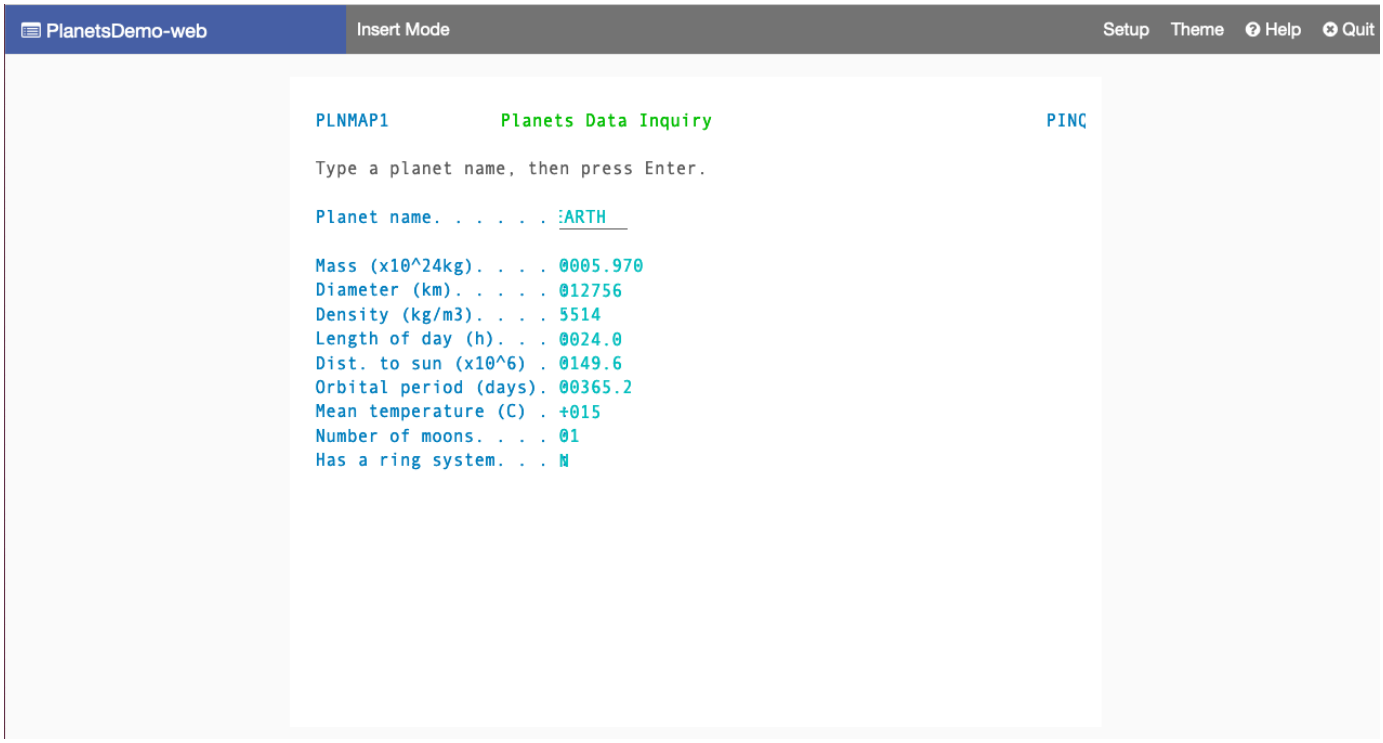



4. Um die Anwendungstransaktion zu starten, geben Sie den `PINQ` Text in das Eingabefeld ein und wählen Sie Ausführen (oder drücken Sie die Eingabetaste).

   Der Bildschirm der Demo-App sollte erscheinen.



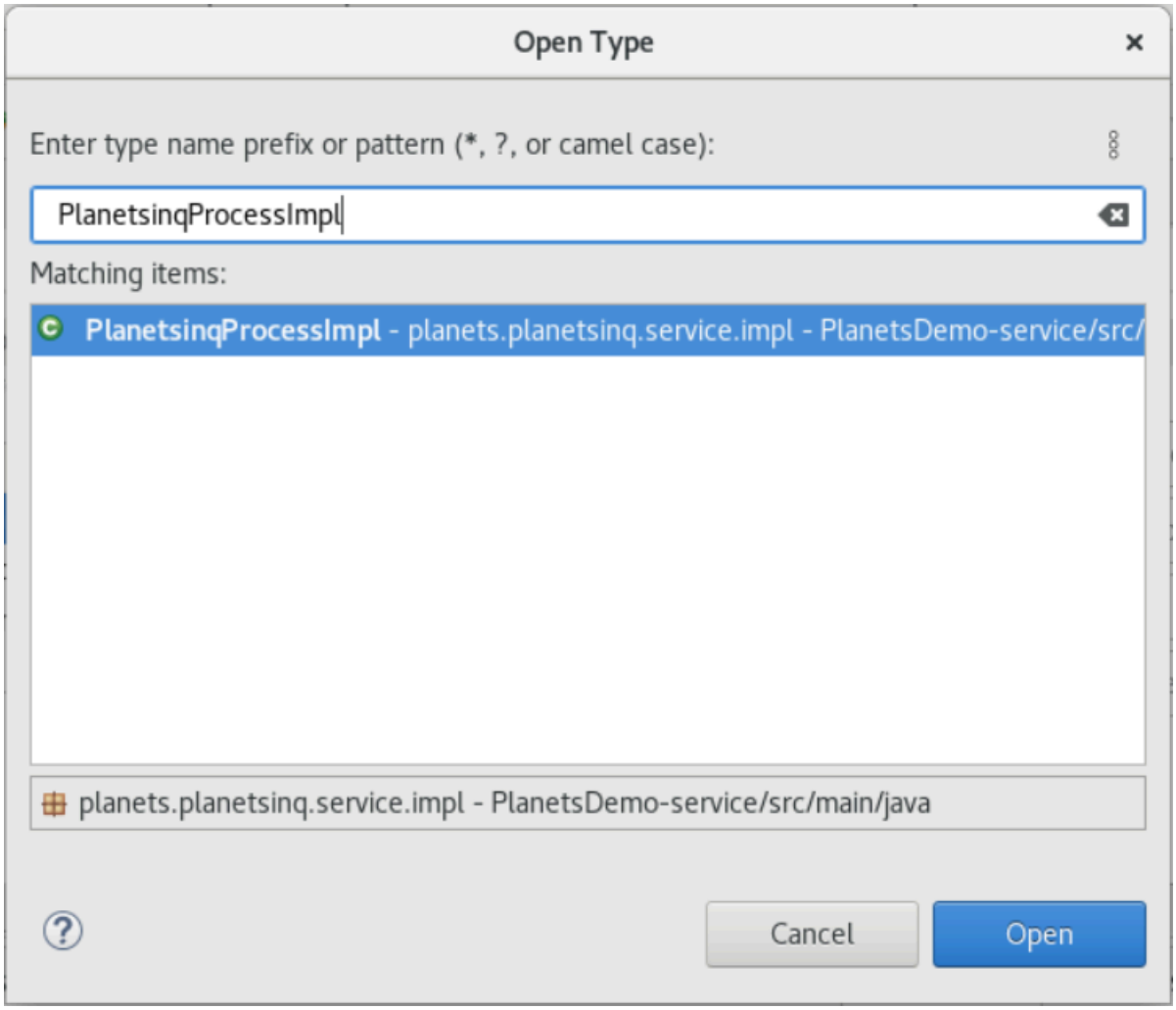

5. Geben Sie einen Planetennamen in das entsprechende Feld ein und drücken Sie die Eingabetaste.



## 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 habenSchritt 1: Erstellen einer Datenbank. Weitere Informationen finden Sie unter Löschen einer DB-Instance.

# AWS Blu Age Häufig gestellte Fragen

## Allgemeines

1. Was ist der Hauptzweck der AWS Refactoring-Funktion von Blu Age?

   Die Refactoring-Funktion refaktoriert älteren monolithischen Code in Java, wobei moderne verteilte Anwendungen verwendet werden, die moderne Sprachen und Frameworks verwenden und dabei einem automatisierten Refactoring-Muster folgen. Dieses Muster beinhaltet die automatische Analyse von Legacy-Code, das Verständnis seiner Funktionalität und die Konvertierung in gleichwertigen modernen Code unter Beibehaltung der Geschäftslogik. Der Prozess umfasst die Modernisierung nicht nur des Codes, sondern auch des gesamten Anwendungsstapels, der Abhängigkeiten und der Infrastruktur mithilfe automatisierter Tools und Prozesse. Die Lösung zielt darauf ab, die Modernisierung zu beschleunigen und gleichzeitig die funktionale Gleichwertigkeit und Leistung aufrechtzuerhalten. Dazu gehört die Transformation sowohl des Anwendungscodes als auch der zugehörigen Datenbanken und Datenspeicher sowie die Implementierung von Best Practices und Entwurfsmustern für die Cloud.

2. Welche Mainframe-Anwendungen werden von AWS Blu Age unterstützt?

   AWS Blu Age unterstützt derzeit die Modernisierung von IBM z/OS Mainframe-Anwendungen, die in COBOL, PL/I, JCL (Job Control Language) geschrieben sind und auf dem CICS-Transaktionsmanager (Customer Information Control System), BMS-Bildschirmen (Basic Mapping Support), IMS-MFS-Screens, Datenbanken, DB2 IMS-Datenbanken, Flatfiles, GDG- (Generation Data Groups) und VSAM-Datendateien (Virtual Storage Access Method) basieren. Weitere Informationen finden Sie unter Blu Insights.AWS

3. Welche Mainframe-Sprachen kann AWS Blu Age modernisieren?

AWS Blu Age transformiert COBOL- und PL/I-Code in Java, in Groovy, JCLs Bildschirme (BMS oder MFS) in HTML (mit Sass) und JavaScript (Angular-Anwendungen — React wird derzeit nicht unterstützt) und ermöglicht so die Modernisierung älterer Mainframe-Anwendungen auf Cloud-native Architekturen. Diese Technologien wurden aufgrund ihrer breiten Akzeptanz, ihres robusten Ökosystems und ihrer Cloud-nativen Funktionen ausgewählt. Angular bietet eine moderne, reaktionsschnelle Benutzeroberflächenebene, die ältere Greenscreen-Benutzeroberflächen ersetzt. Es ermöglicht die Erstellung dynamischer, benutzerfreundlicher Webanwendungen, auf die über verschiedene Geräte und Plattformen zugegriffen werden kann. Die komponentenbasierte Architektur unterstützt eine wartbare und skalierbare Frontend-Entwicklung. Die Transformation führt zu verteilten Anwendungen, die modernen Architekturmustern und bewährten Verfahren folgen.

4. Wie bringt AWS Blu Age bestehende Einschränkungen mit Cloud-Vorteilen in Einklang?

   AWS Blu Age schafft ein Gleichgewicht, indem es wichtige Geschäftslogik und Funktionen beibehält und gleichzeitig Cloud-native Funktionen einführt. Es stellt sicher, dass modernisierte Anwendungen die erforderliche veraltete Geschäftslogik beibehalten und gleichzeitig die Vorteile der Cloud-Skalierbarkeit, Agilität und modernen Betriebspraktiken nutzen. Dieser Ansatz hilft Unternehmen dabei, die Geschäftskontinuität aufrechtzuerhalten und gleichzeitig die Vorteile der Cloud-Infrastruktur zu nutzen.

5. Welche Rolle spielt die serviceorientierte Architektur in der modernisierten Anwendung?

   Die serviceorientierte Architektur spielt eine grundlegende Rolle bei der Aufteilung monolithischer Anwendungen in besser verwaltbare modulare Komponenten. AWS Blu Age entwickelt serviceorientierte und objektorientierte Anwendungen, die eine bessere Wartbarkeit und Skalierbarkeit ermöglichen. Dieser architektonische Ansatz ermöglicht es Unternehmen, ihre Geschäftseffizienz zu steigern und sich auf eine mögliche future Einführung von Microservices vorzubereiten.

6. Welche Aspekte des Anwendungsstapels sind im Refactoring-Prozess enthalten?

   Der Refactoring-Prozess umfasst den kompletten Software-Stack: Anwendungscode, Abhängigkeiten, Datenbanken und Infrastruktur (z. B. Optionen für Caching, Messaging-Unterstützung usw.). Es umfasst die Transformation älterer Programmiersprachen, Datenbanksysteme, Datendateien und der zugehörigen Infrastrukturkomponenten. Dieser umfassende Ansatz stellt sicher, dass alle Aspekte der Anwendung einheitlich modernisiert werden, was zu einem vollständig transformierten modernen Anwendungsstapel führt.

7. Macht der Modernisierungsprozess von AWS Blu Age jegliche Tests oder Qualitätssicherungsprüfungen der modernisierten Java-Anwendung überflüssig?

   Nein, der Modernisierungsprozess von AWS Blu Age macht Tests oder Qualitätssicherungsprüfungen der modernisierten Java-Anwendung nicht überflüssig.

8. Wofür steht AWS Blu Age JAC?

   JAC steht für JICS Administration Console

9. Wie kann ich auf die AWS Blu Age-Tools zugreifen?

   AWS Auf Blu Age-Tools kann über die AWS-Konsole über AWS Mainframe Modernization (M2) Refactor zugegriffen werden. Der Zugriff auf Funktionen hängt von Ihrer Akkreditierungsstufe ab. Beginnen Sie mit dem Transformation Center, um das automatische Java-Refactoring Ihres Quellcodes zu bewerten. Eine ausführliche Anleitung finden Sie in der AWS Blu Insights-Dokumentation. Nach der Modernisierung können Sie Anwendungen entweder mit verwalteten oder nicht verwalteten Laufzeitoptionen bereitstellen. Weitere Informationen zu diesen Bereitstellungsoptionen finden Sie in der Dokumentation zur AWS Mainframe-Modernisierung.

10. Wie berechnet man die Größe (Arbeitslast und Zeitplan) eines Projekts?

    Weitere Informationen dazu finden Sie unter Schätzungen von AWS Blu Insights oder wenden Sie sich an Ihren Account Manager.

11. Gibt es spezielle Anforderungen für die Wartung von migrierten Java AWS Blu Age-Lösungen?

    Nein, es gibt keine spezifischen Anforderungen für die Wartung von mit Java AWS Blu Age migrierten Lösungen.

12. Was sind die technischen Spezifikationen und die Kompatibilität des von AWS Blu Age generierten Codes?

    AWS Der von Blu Age generierte Code wurde mit spezifischen technischen Eigenschaften und umfassender Kompatibilität entwickelt. JPA wird zwar nicht unterstützt, verwendet jedoch die direkte SQL-Ausführung mit externalisierten Abfragen. Der Code stützt sich auf laufzeitspezifische Bibliotheken für funktionale Äquivalenz, Generierung von Webdiensten und MQ-Implementierungen. Der generierte Code kann zur Entwicklung, zum Testen, Erstellen und Bereitstellen in jede Java-IDE importiert werden. Die erforderlichen Bibliotheken müssen jedoch entsprechend importiert werden. Während Maven standardmäßig in den AWS Mainframe Modernization Service für Build-Prozesse integriert ist, können alternative Tools wie Gradle verwendet werden, indem das Paketierungsformat nach der Transformation geändert wird.

Die Plattform bietet Flexibilität in Bezug auf Entwicklungstools und Quellcodeverwaltung. Für Entwicklungsteams, die den Code verwalten, stehen Schulungen zur Verfügung. Weitere Informationen finden Sie unter AWS Blu Age Runtime High-Level-Architektur.

## AWS Blu Age Runtime

1. Wo finde ich Informationen zu AWS Blu Age Runtime?

   Informationen zur nicht verwalteten Laufzeit finden Sie in der Dokumentation zur Einrichtung von AWS Blu Age Runtime (nicht verwaltet), in der der Einrichtungsprozess, das Onboarding, das Abrufen von Artefakten, die Bereitstellung usw. beschrieben werden.

2. Wo finde ich AWS Blu Age Runtime für Entwickler?

   Die AWS Blu Age Runtime für Entwickler ist in der Blu Age Toolbox für L3-zertifizierte Personen verfügbar.

3. Werden die AWS Blu Age-JAR-Abhängigkeiten zur lokalen Entwicklung in das Maven-Repository des Kunden hochgeladen?

   Bibliotheken können EC2 mithilfe eines AMI importiert werden, das für die Konfiguration der Entwicklungs-, Test- und Produktionsumgebung verwendet werden kann. Das Team wird geschult und unterstützt, um den generierten Anwendungscode zu warten/zu verbessern. Weitere Informationen finden Sie unter AWS Blu Age Runtime High-Level-Architektur.

4. Worauf bezieht sich der Begriff „Gapwalk“ in den verteilten AWS Blu Age Runtime-Jars?

   Informationen zu Gapwalk finden Sie unter AWS Blu Age Runtime-Artefakte.

5. Wie beantrage ich Zugriff auf die nicht verwaltete AWS Blu Age Runtime?

   Folgen Sie den Anweisungen unter Onboarding von AWS Blu Age Runtime, um Zugang zum Center zu AWS -Support beantragen.

6. Was sind die unterstützten Runtimes für umgestaltete AWS Blu Age-Anwendungen?

   Um die gesamte Palette der Runtime-Optionen für Ihre modernisierten Anwendungen zu erkunden, empfehlen wir Ihnen, den Leitfaden zu den Blu Age-Runtime-Optionen zu lesen.

7. Wann wird AWS Blu Age Runtime verwendet?

   Eine AWS Blu Age Runtime ist erforderlich, um die Ausführung von umgestalteten AWS Blu Age-Anwendungen zu unterstützen. Bei AWS Blu Age-basierten Refactoring-Projekten ist eine

Runtime erforderlich, um die umgestalteten Anwendungen zu testen. Sobald das Refactoring-Projekt abgeschlossen ist, wird auch eine Runtime für die Wartung, das Testen und den Betrieb von umgestalteten Blu-Age-Anwendungen in der Produktion benötigt. AWS

8. Wie AWS verteilt man neue Releases für Blu Age Runtime? AWS

   Für M2 Managed Runtime werden Updates, einschließlich Patches, Neben- und Hauptversionen, in der AWS Konsole und verfügbar gemacht AWS CLI. Sie beinhalten Betriebssystem-Updates, Engine- und Abhängigkeitsänderungen, die in der Regel innerhalb von 30 Tagen nach der allgemeinen Verfügbarkeit erfolgen. AWS ist für die unterstützten Komponenten verantwortlich und wendet Updates automatisch auf AWS Mainframe-Modernisierungsinstanzen an. Und das Gleiche gilt für andere Umgebungen wie Custom Runtime, Linux AMI und lokale Umgebungen.

9. Wie oft werden neue Haupt- und Nebenversionen von AWS Blu Age Runtime veröffentlicht?

   Neue Versionen werden ein bis zwei Monate veröffentlicht, und Kunden können entscheiden, wann und wie sie ihre Runtime-Instances aktualisieren möchten. Weitere Informationen finden Sie auf der Seite AWS Blu Age-Versionierung.

10. Wie bietet AWS Support für AWS Blu Age Runtime?

    Der Support erfolgt über AWS -Support, bei dem Probleme behoben werden, indem ein Ticket erstellt wird und die Standard-SLA gilt. Weitere Informationen finden Sie unter Lebenszyklus der AWS Mainframe-Modernisierungskomponenten.

11. Was beinhaltet die AWS Blu-Age-Runtime für die AWS Mainframe-Modernisierung?

    Die AWS Blu Age Runtime umfasst Toolbox-Bibliotheken zur Beschleunigung der Modernisierung, zur Erleichterung von Cloud-Integrationen und zur Verbesserung der Codequalität und Wartbarkeit. Sie ermöglicht auch eine stärkere Automatisierung der Modernisierung, indem sie den Übergang zwischen Legacy-Architekturen und Cloud-Architekturen erleichtert. Die Runtime bietet Unterstützung für den Umgang mit Speicherdarstellungen älterer Verben und Datenstrukturen mithilfe von Java-Idiomen. Sie ermöglicht die Erstellung modernisierter Anwendungen, die auf objektorientierten Programmiertechniken basieren und in der Lage sind, ältere Kontrollabläufe zu reproduzieren. Es modernisiert ältere VSAM-Datensätze oder die Unterstützung hierarchischer IMS-Datenbanken mithilfe einer relationalen Datenbank wie Amazon Aurora. Es bietet Java-Ersatz für ältere Systemdienstprogramme (IDCAMS, IEBGENER, DFSORT usw.) und ältere Transaktionsmanagementsysteme (CICS, IMS). Es erleichtert Cloud-Integrationen mit Caching in Amazon ElastiCache und Unterstützung für AWS Messaging-Lösungen (SQS, Kinesis).

12. Unterstützt AWS Blu Age Runtime Computerarchitekturen, die nicht auf x86 basieren?

Derzeit unterstützt AWS Blu Age Runtime nur x86-basierte Computerarchitekturen und Rechenleistung. AWS Blu Age Runtime unterstützt keine ARM- und Graviton-basierten Computer.

13. Wie können Kunden über AWS Blu Age Runtime-Versionen auf dem Laufenden bleiben, einschließlich Benachrichtigungen über neue Versionen und Zugriff auf Versionsverlauf und Versionshinweise?

    Neue Versionen von AWS Blu Age Runtime werden auf unsere offizielle Release-Seite hochgeladen. Wir empfehlen, diese Seite regelmäßig, idealerweise alle 3 Monate, auf die neuesten Versionen und Updates zu überprüfen. Was den Zugriff auf den Versionsverlauf und die Versionshinweise angeht, hängt die Verfügbarkeit vom end-of-life (EOL-) Datum der jeweiligen Hauptversion ab. Ausführliche Informationen zu EOL-Daten, zur Planung von Versionsupgrades und zum Zugriff auf historische Informationen finden Sie unter AWS Blu Age-Lebenszyklus.

14. Was sind die Hauptkomponenten der High-Level-Architektur von AWS Blu Age Runtime?

    Die AWS Blu Age Runtime-Architektur besteht aus zwei Hauptkomponententypen. Erstens handelt es sich um Java-Bibliotheken (JAR-Dateien), die in einem gemeinsam genutzten Ordner gespeichert sind (auf den der Klassenlader des Anwendungsservers zugreifen kann) und Unterstützung für ältere Konstrukte und Anweisungen bieten. Zweitens gibt es Webanwendungen (Kriegsdateien), die Spring-basierte Anwendungen enthalten, die Frameworks und Dienste für modernisierte Programme bereitstellen. Zur Laufzeit gehören außerdem: eine Programmregistrierung, in der alle Programme für Aufrufe und programmübergreifende Aufrufe gesammelt werden, und eine Skriptregistrierung, die alle modernisierten Jobskripte sammelt. Diese Komponenten arbeiten zusammen, um ein einheitliches REST-basiertes Einstiegs- und Ausführungsframework für modernisierte Anwendungen bereitzustellen. Die Runtime und die modernisierte Anwendung werden zusammen auf einem Anwendungsserver (z. B. Tomcat) bereitgestellt.

15. Wie konfiguriere ich den gemeinsamen Ordner mit AWS Blu Age Runtime-Artefakten?

    Die AWS Blu Age Runtime-Artefakte (Jars) müssen in einem gemeinsamen Ordner gesammelt werden, auf den der Klassenlader des Anwendungsservers zugreifen kann. Bei einem Tomcat-Server erfolgt die Konfiguration durch Ändern der regulären Konfigurationsdatei mit dem Namen catalina.properties. Wenn Sie den gemeinsamen Ordner beispielsweise als Ordner mit dem Namen „shared“ im Tomcat-Ordner erstellt haben, müssen Sie den Eintrag common.loader in der Datei catalina.properties ändern, um den gemeinsamen Ordner für den Tomcat-Klassenlader wie folgt zugänglich zu machen:

```
common.loader="${catalina.base}/lib","${catalina.base}/lib/*.jar","${catalina.home}/
lib","${catalina.home}/lib/*.jar","${catalina.home}/shared","${catalina.home}/shared/
*.jar"
```

16. Wie geht AWS Blu Age Runtime mit Staatenlosigkeit und Sitzungsmanagement um?

    AWS Blu Age Runtime implementiert Zustandslosigkeit und Sitzungsmanagement über mehrere Mechanismen. Für HTTP-Sitzungen verwendet es eine Cookie-basierte Identifizierung mit externem Cache-Speicher für den Benutzerkontext. Sitzungen können in verschiedenen Datenspeichern gespeichert werden, darunter Amazon ElastiCache, Redis-Cluster oder In-Memory-Maps. Das Statelless-Design stellt sicher, dass die meisten nicht vorübergehenden Zustände extern in einer gemeinsamen „zentralen Informationsquelle“ gespeichert werden, was eine hohe Verfügbarkeit und horizontale Skalierung ermöglicht. Dieser Ansatz ermöglicht in Kombination mit Lastenausgleich und gemeinsam genutzten Sitzungen die Verteilung von Benutzerdialogen auf mehrere Knoten.

17. Welche Rolle spielen Webanwendungen in der AWS Blu Age Runtime-Umgebung?

    Webanwendungen in AWS Blu Age Runtime erfüllen mehrere wichtige Funktionen. Sie bieten Ausführungs-Frameworks, die ältere Umgebungen und Transaktionsmonitore (wie JCL-Batches, CICS, IMS) reproduzieren. Sie bieten REST-basierte Einstiegspunkte `gapwalk-application.war` zum Auslösen und Steuern von Transaktionen, Programmen und Batches. Darüber hinaus ermöglichen sie die Emulation von vom Betriebssystem bereitgestellten Programmen und speziellen Treiberprogrammen, auf die ältere Anwendungen angewiesen sind, um über MFS auf Dienste wie IMS DB oder Benutzerdialoge zuzugreifen.

18. Wie werden Programme in Blu Age Runtime registriert und verwaltet? AWS

    Programme in AWS Blu Age Runtime werden über ein ProgramRegistry System registriert, das beim Serverstart automatisch geladen wird. Jedes Programm implementiert die Programmoberfläche und ist als Spring-Komponente gekennzeichnet. Programme werden anhand ihrer Bezeichner registriert, wobei mehrere Einträge möglich sind, wenn ein Programm mehrere Bezeichner hat. Der Registrierungsprozess erfolgt automatisch und wird in Tomcat-Protokollen protokolliert. ProgramRegistryDies ermöglicht es anderen Programmen und Skripten, registrierte Programme zu finden und aufzurufen, wobei die Modularität und Interkonnektivität des modernisierten Systems erhalten bleibt.

19. Wie wird die Konfiguration in AWS Blu Age Runtime-Anwendungen verwaltet?

Die Konfiguration in AWS Blu Age Runtime wird mithilfe von YAML-Dateien mithilfe der Funktionen des Spring Boot-Frameworks verwaltet. Zwei Hauptkonfigurationsdateien werden verwendet: application-main.yml für die Framework-Konfiguration und für kundenspezifische Optionen. `application-profile.yml` Das System folgt der Prioritätslogik von Spring und ermöglicht das Überschreiben der Konfiguration auf verschiedene Weise. Zusätzliche Konfigurationen können über JNDI für Datenbanken und Befehlszeilenparameter bereitgestellt werden, was Flexibilität bei der Konfigurationsverwaltung bietet. Die Logger-Konfiguration erfolgt mithilfe von Logback-XML-Konfigurationsdateien.

20. Welche Rolle spielen Secrets Manager in der AWS Blu Age Runtime-Konfiguration?

    Secrets-Manager in AWS Blu Age Runtime schützen sensible Konfigurationsdaten wie Datenbankanmeldedaten und Redis-Cache-Passwörter. Sie ermöglichen die Speicherung kritischer Daten in AWS Geheimnissen und referenzieren sie in YAML-Konfigurationsdateien. Das System unterstützt verschiedene Arten von Geheimnissen, darunter Datenbankgeheimnisse, die automatisch alle relevanten Felder ausfüllen, und Geheimnisse mit einem einzigen Passwort für kennwortgeschützte Ressourcen. Dieser Ansatz erhöht die Sicherheit, indem sensible Daten von der Anwendungskonfiguration getrennt werden.

21. Wie können Entwickler ihre eigenen Programme schreiben, die mit AWS Blu Age Runtime kompatibel sind?

    Entwickler können mit AWS Blu Age Runtime kompatible Programme erstellen, indem sie die Programmoberfläche implementieren und bestimmten Mustern folgen. Das Programm muss als Spring-Komponente deklariert werden, die erforderlichen Methoden implementieren und ordnungsgemäß in der registriert sein. ProgramRegistry Entwickler müssen Begleitkontext- und Konfigurationsklassen erstellen, Programmbezeichner verwalten und die korrekte Integration in das Spring-Framework sicherstellen. Die Implementierung sollte den Konventionen von AWS Blu Age Runtime in Bezug auf Programmstruktur und Ausführung entsprechen.

22. Wie geht AWS Blu Age Runtime mit Fehlern bei der Programmausführung um?

    AWS Blu Age Runtime behandelt Fehler bei der Programmausführung über mehrere Mechanismen. Bei Batch-Jobs werden der Ausführungsstatus, die Exit-Codes und detaillierte Fehlerinformationen in den Details zur Jobausführung erfasst. Die Fehlerbehandlung umfasst spezifische Exit-Codes (-1 für technische Fehler, -2 für Ausfälle von Serviceprogrammen) und eine detaillierte Protokollierung in Tomcat-Protokollen. Das System kann so konfiguriert werden, dass Transaktionen bei Laufzeitausnahmen rückgängig gemacht werden, und bietet Optionen für

die Fehlerbenachrichtigung und Wiederherstellung. Fehlerdetails sind über REST-Endpunkte zur Überwachung und Fehlerbehebung zugänglich.

23. Welche Überwachungsfunktionen von AWS Blu Age Runtime sind für Batch-Jobs verfügbar?

AWS Blu Age Runtime bietet Überwachungsfunktionen für Batch-Jobs über verschiedene Endpunkte. Es verfolgt den Status der Auftragsausführung, die Start-/Endzeiten, den Ausführungsmodus und detaillierte Ergebnisse. Das System bietet Endpunkte zum Auflisten von ausgelösten Skripten, zum Abrufen von Details zur Auftragsausführung und zum Überwachen aktuell ausgeführter Jobs. Die Endgeräte von Metrics bieten JVM-Statistiken, Sitzungszählungen und detaillierte Metriken zur Batch-Ausführung. Die Plattform unterstützt auch Paginierung und zeitbasiertes Filtern von Überwachungsdaten.

24. Wie wird der Ausführungsstatus von AWS Blu Age Runtime-Jobs verfolgt und verwaltet?

Der Status der Auftragsausführung wird über ein umfassendes Statussystem verfolgt, das Status wie DONE, TRIGGERED, RUNNING, KILLED und FAILED umfasst. Jede Auftragsausführung erhält eine eindeutige Kennung für die Nachverfolgung und verwaltet detaillierte Ausführungsinformationen wie Startzeit, Endzeit, Anruferinformationen und Ausführungsergebnisse. Das System stellt REST-Endpunkte zur Abfrage des Auftragsstatus, zur Verwaltung laufender Jobs und zum Abrufen des Ausführungsverlaufs bereit. Statusinformationen verbleiben im Serverspeicher und können je nach Alter für die Ressourcenverwaltung gelöscht werden.

25. Wie geht AWS Blu Age Runtime mit externen Systeminteraktionen um?

Die Runtime verarbeitet externe Systeminteraktionen über verschiedene Mechanismen, darunter REST-Endpunkte für die Serviceintegration, Unterstützung für Nachrichtenwarteschlangen (SQS, RabbitMQ, IBM MQ) und Datenbankverbindungsoptionen. Sie ermöglicht die Emulation älterer Systeminteraktionen durch spezielle Komponenten, unterstützt SSL/TLS für sichere Kommunikation und umfasst Funktionen für den Umgang mit externen Dateisystemen. Das System unterstützt auch die Integration mit externen Authentifizierungsanbietern und kann für die Interaktion mit verschiedenen Diensten von Drittanbietern konfiguriert werden.

26. Wie wird die Authentifizierung in AWS Blu Age Runtime gehandhabt?

AWS Blu Age Runtime unterstützt mehrere Authentifizierungsmethoden, OAuth2 wobei es sich um den primären Mechanismus handelt. Es kann in Identitätsanbieter wie Amazon Cognito oder Keycloak integriert werden. Die Authentifizierungskonfiguration wird über die Hauptkonfigurationsdatei namens application-main.yml verwaltet, in der Sicherheitseinstellungen, Identitätsanbieter und Authentifizierungsmethoden definiert werden können. Das System

unterstützt Funktionen wie XSS-Schutz, CORS und CSRF und kann sowohl für globale Sicherheit als auch für spezifische Endpunktsicherheit konfiguriert werden. Für die Entwicklung steht auch ein lokales Authentifizierungssystem mit standardmäßigen Super-Admin-Anmeldeinformationen zur Verfügung.

27. Wie gewährleistet AWS Blu Age Runtime eine hohe Verfügbarkeit?

    AWS Blu Age Runtime gewährleistet Hochverfügbarkeit durch verschiedene Mechanismen. Es implementiert Zustandslosigkeit, indem nichttransiente Zustände in einem externen, gemeinsam genutzten Speicher gespeichert werden, sodass mehrere Anwendungsinstanzen zusammenarbeiten können. Das System unterstützt Lastenausgleich und gemeinsame Sitzungen, sodass Anfragen auf mehrere Knoten verteilt werden können. Für die Datenspeicherung kann es hochverfügbare Datenbanken und Caching-Systeme verwenden. Die Architektur unterstützt automatisches Failover und kann zur Erhöhung der Zuverlässigkeit in mehreren Availability Zones eingesetzt werden.

28. Welche Komponente wird verwendet, um verteilte CICS-Transaktionen mit AWS Blu Age-Anwendungen zu reproduzieren?

    Die AWS Blu Age Runtime bietet einen dedizierten Endpunkt, über den bestehende JICS-Transaktionen als Teil einer globalen Transaktion aufgerufen werden können (XA-Unterstützung). Die zugrundeliegenden zwei Phasen der Commit-Unterstützung basieren auf der Atomikos-Softwarekomponente.

29. Wie lautet der AWS Blu-Age-Name der Klassen, die zur Definition eines bestimmten Programmverhaltens verwendet werden?

    Jedes Programm ist an eine spezielle Konfigurationsklasse gebunden, die es ermöglicht, das programmspezifische Verhalten zu spezifizieren. Weitere Informationen zu Benennungs- und Standortkonventionen finden Sie unter AWS Blu Age-Struktur der modernisierten Anwendung

30. Welche Kodierung hat die folgende Reihenfolge der Zeichen: Leerzeichen, Kleinbuchstaben, Großbuchstaben, Zahlen?

    Zeichensätze, die zur EBCDIC-Variantenfamilie gehören (z. B. CP1 047 usw.). CP297

31. Wie bedient man AWS Blu Age Managed Runtime?

    Mit dem AWS Management Console AWS CLI, dem oder dem AWS APIs

32. Was sind die Preisdimensionen für AWS Blu Age Runtime?

    AWS Mainframe Modernization-core-hours (siehe Preise für AWS Mainframe-Modernisierung).

33. Welcher Mechanismus wird verwendet, um Rohdaten über HTTP an die Programmendpunkte weiterzuleiten?

    Base64-kodierte Zeichenketten.

34. Wie startet ein Benutzer eine Batch-Job-Ausführung?

    Verwenden eines HTTP-Aufrufs an einen der dedizierten Batch-Endpunkte (siehe Dokumentationsseite zu Batch-Endpunkten).

35. Welcher AWS Blu Age Runtime-Endpunkt ist der Haupteingangspunkt der Haupt-Web-Frontend-Anwendung?

    ```
    /transaction
    ```

36. Wofür steht AWS Blu Age JICS?

    Das AWS Blu Age JICS ist die Laufzeitkomponente, die zur Unterstützung der Modernisierung von CICS-Ressourcen verwendet wird. Die Ressourcendefinitionen werden in einem speziellen Datenspeicher gespeichert. Um sie zu verwalten, verwenden Sie entweder die REST-API oder die JICS-Anwendungskonsole. Weitere Informationen finden Sie unter JICS-Anwendungskonsole verwalten in AWS Blu Age.

37. Welche Caching-Mechanismen von AWS Blu Age Runtime sind verfügbar?

    AWS Blu Age Runtime unterstützt mehrere Caching-Mechanismen, darunter Redis und. EhCache Redis wird für Produktionsumgebungen empfohlen, da es ein gemeinsames persistentes Caching über mehrere Knoten hinweg ermöglicht. EhCache ist für eigenständige Bereitstellungen mit eingebettetem volatilem lokalem Caching verfügbar. Das System unterstützt das Zwischenspeichern verschiedener Komponenten, darunter Blusam-Daten, Sitzungsinformationen, JICS-Ressourcen und temporäre Speicherwarteschlangen. Die Cache-Konfiguration kann an unterschiedliche Anwendungsfälle und Leistungsanforderungen angepasst werden.

38. Wie schätzen wir den Preis einer AWS Blu-Age Runtime-Implementierung für die AWS Mainframe-Modernisierung ein?

    AWS bietet Kunden Schätzungen auf der Grundlage ihrer Anforderungen und ihrer Zielarchitektur.

39. Wie hoch ist der Preis für AWS Mainframe Modernization AWS Blu Age Runtime?

    AWS Mainframe Modernization bietet zwei Preismodelle für AWS Blu Age: eine Managed Runtime-Option, die Laufzeit, Rechenressourcen, internen Speicher und Automatisierung umfasst, und eine Option ohne verwaltete Runtime, die nur die AWS Blu Age Age-Laufzeit selbst abdeckt.

Für AWS Bereitstellungen verwenden beide eine pay-as-you-go Preisstruktur. Die ausführlichsten up-to-date Preisinformationen finden Sie auf der offiziellen Seite mit den Preisen für AWS Mainframe Modernization.

40. Was ist, wenn wir eine umgestaltete AWS Blu Age-Anwendung auf einer Infrastruktur bereitstellen müssen, die nicht in der unterstützten Runtime aufgeführt ist?

    Wenn Sie eine umgestaltete AWS Blu Age-Anwendung auf einer Infrastruktur bereitstellen müssen, die nicht in der unterstützten Runtime aufgeführt ist, stehen Ihnen mehrere Optionen zur Verfügung. Prüfen Sie zunächst, ob Ihre Infrastruktur mit bestehenden Bereitstellungsoptionen wie Amazon EKS Anywhere oder anderen Container-Orchestrierungsplattformen kompatibel ist. In diesem Fall können Sie möglicherweise die AWS Blu Age Runtime (nicht verwaltet) verwenden. Bei nicht kompatiblen Infrastrukturen empfehlen wir, sich mit einem AWS Mainframe-Spezialisten in Verbindung zu setzen, um maßgeschneiderte Lösungen oder mögliche Anpassungen zu prüfen. Sie können auch einen Product Feature Request (PFR) für erweiterten Infrastruktursupport einreichen. Für nicht standardmäßige Bereitstellungen sind möglicherweise alternative Abrechnungsoptionen verfügbar. Wenden Sie sich an Ihren AWS Vertreter, um Ihre spezifischen Anforderungen und den besten Ansatz für Ihre Umgebung zu besprechen.

41. Wie wird AWS Blu Age Runtime lizenziert? Ist sie Open Source?

    AWS Blu Age Runtime ist kein Open-Source-Programm. Es wird über AWS IP als Cloud-nativer Dienst vertrieben. Es gibt zwei Bereitstellungsoptionen:

    a. AWS Blu Age Managed, die Runtime wird in einem speziellen AWS Managed Service bereitgestellt, der die Vorteile einer vollständig vorkonfigurierten und sofort einsatzbereiten Umgebung ohne Einrichtung oder Administration nutzt.

    b. AWS Blu Age Non Managed, das in Ihrer eigenen maßgeschneiderten AWS Architektur auf Basis von Amazon EC2 oder Amazon ECS/ bereitgestellt werden kann AWS Fargate, die Sie selbst bereitstellen und einrichten müssen. Für beide Optionen fallen Laufzeitgebühren an, die in den Ihnen zur Verfügung gestellten Projektschätzungen enthalten sind. Da es sich um einen verwalteten Dienst mit Support Zugriff handelt, benötigen Sie den Quellcode nicht. Weitere Informationen zur Preisgestaltung finden Sie auf der Seite mit den Preisen für die AWS Mainframe-Modernisierung.

42. Wie werden Änderungen und Upgrades an AWS Blu Age-Frameworks und -Bibliotheken verwaltet?

    AWS Die Frameworks und Bibliotheken von Blu Age werden im Rahmen regelmäßiger Codegenerierungs- und Bereitstellungsprozesse aktualisiert. Diese Updates werden im Rahmen

des AWS Mainframe-Modernisierungszyklus verwaltet, der Versionsupgrades und Support durch das AWS Blu Age Age-Team oder zertifizierte Partner umfasst. Detaillierte Informationen zu Versionierung, Upgrade-Prozessen und Supportzeitplänen finden Sie in der Dokumentation zum Lebenszyklus der AWS Mainframe-Modernisierung.

## Daten

1. Welche Datenbankoptionen sind für die modernisierten Anwendungen in Bezug auf die Modernisierung der Legacy-Datenbank verfügbar?

   Die modernisierten Anwendungen können mehrere moderne Datenbankoptionen verwenden, darunter: PostgreSQL, Amazon Aurora, RDS für PostgreSQL, Oracle-Datenbank, MS-SQL und IBM Db2. Diese Optionen bieten Flexibilität bei der Auswahl des am besten geeigneten Datenbanksystems auf der Grundlage spezifischer Anforderungen und nutzen gleichzeitig die Vorteile moderner Datenbankmanagementsysteme und Cloud-nativer Funktionen.

2. Wozu dient der Transformationsbereich von IBM Db2 z/OS zu Postgres DDL?

   Vollständige Transformation (einschließlich Datenbankeinschränkungen).

3. Unterstützt AWS Blu Age die Gruppendatengenerierung (GDG)?

   Ja, die Verwendung von GDG in Batches wird unterstützt, wobei relative und absolute Generierungen sowie automatische Bereinigungsstrategien unterstützt werden.

4. Unterstützt AWS Blu Age verkettete Datensätze?

   Ja, die Verwendung verketteter Datensätze in Batches wird unterstützt. Wenn die Verkettung aktiviert ist, können mehrere Datensätze als ein einziger Datensatz gelesen werden. Bitte beachten Sie, dass die Blusam-Datensätze nicht Teil einer Verkettung sein können.

5. Welcher Prozess wird auf SQL-Abfragen angewendet?

   Wird je nach Zieldatenbank während der Codetransformation angepasst.

6. Welche Optionen gelten, wenn es mehrere Datenbanken für eine Anwendung gibt?

   Konfigurieren Sie die Zieldatenbank für jede Abfrage und definieren Sie alle Datenbanken in der Anwendung und in Apache Tomcat.

7. Kann Blusam deaktiviert werden?

Ja, in der Hauptkonfigurationsdatei, und es ist keine Datenbank erforderlich (weitere Informationen finden Sie auf der Dokumentationsseite zur Blusam-Konfiguration).

8. Welche AWS Blu Age-API wird verwendet, um Datenbanken wie IMS DB zu ersetzen?

   Die JHDB-API (Java Hierarchical DataBase).

9. Welches AWS Blu Age-Produkt kann verwendet werden, um ältere Daten und Datenbanken auf ein modernes relationales Datenbankmanagementsystem (RDBMS) zu migrieren?

   AWS Blu Age DB-Modernisierungstool (Data Migrator).

10. Was ist AWS Blu Age Data Simplifier und welches Problem löst es bei der Modernisierung?

    Data Simplifier ist eine Kernbibliothek in AWS Blu Age, die sich mit der Herausforderung befasst, mit veralteten Speicherzugriffsmustern in Java umzugehen. Sie bietet Konstrukte zur Unterstützung von Low-Level-Speicherzugriffen, älteren Datentypen (wie zonenweise, gepackt, alphanumerisch) sowie gemischten und klassenbasierten Datentypen APIs, sodass sie für Java-Entwickler zugänglich sind structured/raw memory access that are common in mainframe applications but not natively available in Java. The library exposes these features through familiar Java patterns like getters/setters und gleichzeitig die veralteten Funktionen beibehalten werden.

11. Wie geht AWS Blu Age mit veralteten Speicherlayouts und Datenstrukturen um?

    AWS Blu Age verarbeitet ältere Speicherlayouts über die Record-Schnittstelle, die eine Abstraktion von Byte-Arrays mit fester Größe bietet. Für strukturierte Daten wie COBOL '01-Datenelemente' werden RecordEntityUnterklassen verwendet, die bei der Modernisierung automatisch generiert werden. Diese Klassen behalten die hierarchische Struktur der älteren Daten bei, wobei jedes Element über- und untergeordnete Elemente verfügt. Das System unterstützt sowohl Rohspeicherzugriff als auch strukturierte Zugriffsmuster, wodurch die Flexibilität älterer Systeme erhalten bleibt und gleichzeitig eine moderne Programmierschnittstelle bereitgestellt wird.

12. Wie geht AWS Blu Age mit der Modernisierung von VSAM-Datensätzen um?

    Die Blusam-Komponente unterstützt die Modernisierung der VSAM-Datensätze mit einer speziellen API, Endpunkten und einer Verwaltungs-Webanwendung (BAC: Blusam Administration Console). Blusam stützt sich auf eine relationale Datenbank als Backend (PostgreSQL, entweder mit RDS oder Aurora).

## Transformation

1. Wo finde ich Details zum Transformationsprozess?

   Informationen finden Sie in der Dokumentation zu AWS Blu Insights.

2. Wie lauten die Namen der von AWS Blu Age generierten Module?

   Service, Entitäten, Web und Tools.

3. Warum wurde Java/Spring als eine der Zieltechnologien für AWS Blu Age ausgewählt?

   Java/Spring wurde aufgrund der breiten Akzeptanz, des großen Talentpools und der robusten Unternehmensfunktionen als Zieltechnologie ausgewählt. Das Java-Ökosystem bietet umfangreiche Bibliotheken, Frameworks und Tools, die die moderne Anwendungsentwicklung unterstützen. Das Spring Framework bietet Funktionen der Enterprise-Klasse, Cloud-native Funktionen und folgt den branchenüblichen Best Practices, sodass es sich ideal für modernisierte Anwendungen eignet.

4. Wie lautet der Name des übergeordneten Projekts, das die von AWS Blu Age generierten Module enthält?

   Der Name des übergeordneten Projekts hat das Suffix „-pom“ und kann im Transformation Center mithilfe der Transform-Eigenschaft mit dem Namen project definiert werden.

5. Wie verwaltet AWS Blu Age die Modernisierung älterer Scheduler, falls sie bereitgestellt wird?

   Ältere Scheduler-Assets werden von Blu Age nicht modernisiert AWS . Sie werden in der Bewertungsphase berücksichtigt, um mögliche fehlende Artefakte zu identifizieren.

6. Was ist die Voraussetzung für das Debuggen des generierten Codes mit AWS Blu Age?

   Jede integrierte Entwicklungsumgebung (IDE), die Java unterstützt, wie Eclipse, JetBrain, oder VisualCode.

## Bereitstellung

1. Welche Umgebungen stehen zur Verfügung, um die modernisierte Anwendung mit AWS Blu Age bereitzustellen?

   Windows Server, Linux-Server und Docker-Linux-Container.

2. Können refaktorierte AWS Blu Age-Anwendungen auf jeder Infrastruktur ausgeführt werden?

Refactorierte AWS Blu Age-Anwendungen sind zwar nicht für die Ausführung auf einer beliebigen Infrastruktur konzipiert, bieten jedoch eine erhebliche Flexibilität bei den Bereitstellungsoptionen. Diese Anwendungen können auf verschiedenen Computerplattformen bereitgestellt werden, darunter Cloud-verwaltete Dienste, serverlose Datenverarbeitung und lokale Infrastrukturen. AWS Blu Age bietet sowohl verwaltete als auch nicht verwaltete Runtime-Optionen, sodass Unternehmen je nach ihren spezifischen Bedürfnissen und Anforderungen zwischen vollständig verwaltetem Komfort und maßgeschneiderter Steuerung wählen können. Diese Flexibilität ermöglicht eine einfache Übertragung zwischen unterstützten Infrastrukturen, sodass sich die umgestalteten AWS Blu Age-Anwendungen hervorragend an unterschiedliche Bereitstellungsumgebungen anpassen lassen. Weitere Informationen finden Sie in der Dokumentation zu den AWS Blu Age Runtime-Optionen.

3. Welche MQ-Konfiguration unterstützt AWS Blu Age?

   SQS, IBM WebSphere MQ.

4. Auf welchen Anwendungsservern kann ein Benutzer Java-Geschäftsanwendungslogik mit nicht verwalteter Laufzeit von AWS Mainframe Modernization bereitstellen?

   Apache Tomcat, Version größer oder gleich 10.1.

5. Wie lässt sich die umgestaltete Anwendung in andere Anwendungen AWS-Services wie Amazon Aurora integrieren?

   Die modernisierte Anwendung lässt sich integrieren und AWS-Services unterstützt die Transformation zu Cloud-nativen Datenbanklösungen wie Amazon Aurora und RDS für PostgreSQL. AWS Blu Age gewährleistet die Integration zwischen modernisierten Anwendungen und ermöglicht es Unternehmen AWS-Services, Cloud-Funktionen zu nutzen. Diese Integration erstreckt sich sowohl auf Datenspeicher als auch auf Anwendungsdienste innerhalb des AWS Ökosystems. Neben dem Datenbankspeicher lässt sich AWS Blu Age Runtime auch in verschiedene Anwendungen integrieren, AWS-Services darunter Amazon ElastiCache für Redis-Caching, AWS Secrets Manager für das Konfigurationsmanagement und die AWS Mainframe-Modernisierung für die Bereitstellung. Es unterstützt Amazon EC2, Amazon EKS und ECS, die von Fargate für den Containereinsatz verwaltet werden. Das System kann AWS Identity and Access Management für die Authentifizierung Amazon Simple Storage Service für die Speicherung verwenden und unterstützt die Integration mit anderen AWS-Services über Konfiguration und Service Connectors.

6. Wie stellt die umgestaltete Anwendung sicher, dass die Skalierbarkeitsanforderungen erfüllt werden?

Die Lösung gewährleistet Skalierbarkeit, indem sie Anwendungen in Cloud-native Architekturen umwandelt, die die elastische Infrastruktur nutzen können. AWS Sie implementiert moderne Entwurfsmuster und bewährte Verfahren, die eine horizontale und vertikale Skalierung ermöglichen. Der serviceorientierte Ansatz ermöglicht eine unabhängige Skalierung von Komponenten. Die modernisierten Anwendungen können die Vorteile der inhärenten Skalierbarkeitsfunktionen von Cloud-Diensten nutzen.

7. Was passiert, nachdem das Refactoring des Quellcodes abgeschlossen ist?

   Nach dem Refactoring des Quellcodes finden zwei Hauptschritte statt. Zunächst wird die umgestaltete Anwendung erstellt. Zweitens wird die Anwendung in AWS Mainframe Modernization AWS Blu Age Runtime bereitgestellt und überwacht. Die Bereitstellung kann entweder in einer AWS verwalteten Umgebung (AWS Mainframe Modernization Managed Runtime) erfolgen, in der die Infrastruktur automatisiert verwaltet wird, oder in Ihrer AWS-Konto (AWS Mainframe Modernization AWS Blu Age Non-Managed Runtime), in der Kunden ihre eigene Infrastruktur verwalten. Die nicht verwaltete Option kann auf verschiedenen Plattformen bereitgestellt werden, darunter Amazon EC2, ECS on EC2 oder auf Fargate, EKS on. EC2

8. Wie kann ich eine mit AWS Blu Age modernisierte Anwendung auf einem benutzerdefinierten Amazon Linux AMI bereitstellen und ausführen, ohne den Managed Service AWS Mainframe Modernization (M2) zu verwenden?

   Dies kann erreicht werden, indem die Anwendung mithilfe von AWS Blu Age Runtime (nicht verwaltet) auf Amazon EC2 bereitgestellt wird. Der Prozess beinhaltet die Erstellung einer Java/Spring-Anwendung mit einer Abhängigkeit von der AWS Blu Age Runtime-Bibliothek und deren Bereitstellung auf einem benutzerdefinierten Amazon Linux-AMI. Detaillierte Anweisungen zu diesem Ansatz finden Sie unter AWS Blu Age Runtime (nicht verwaltet) bei Amazon EC2 einrichten.

9. Ist ein Amazon Machine Image (AMI) verfügbar? Ist ein Docker-Image verfügbar?

   - AMI: Nein, da die Kunden ihre Umgebung nach ihren Wünschen anpassen und einrichten müssen, ist kein AMI verfügbar. Kunden können die AWS Blu-Age-Artefakte abrufen und ihre Instance entsprechend ihren Anforderungen einrichten.
   - Docker-Image: Nein, es ist kein sofort einsatzbereites Docker-Image verfügbar, aber auf der Seite AWS Blu Age Runtime auf Container einrichten wird erklärt, wie Sie Ihr eigenes Docker-Image auf der Grundlage von AWS Blu Age Runtime-Binärdateien erstellen und in einem geeigneten Container-Management-System bereitstellen.

10. Kann der Kunde eine AWS Blu Age-Anwendung paketieren und als Docker-Container ausführen?

Dies ist für M2 Managed Runtime nicht möglich, aber für kundenspezifische Umgebungen, die auf einem Amazon Linux AMI basieren, und für lokale oder andere Cloud-Anbieter.

11. Wie kann ich den Ressourcen-ARN der SQS-Richtlinie ermitteln, der erforderlich ist, um AWS Blu Age nicht verwaltet auszuführen, wenn ich den Umfang reduzieren möchte?

    Wenden Sie sich an das Bereitstellungsteam oder den Technical Account Manager (TAM), um den spezifischen ARN für die SQS-Richtlinienressource für den Betrieb von AWS Blu Age nicht verwaltet mit einer eingeschränkten Richtlinie zu ermitteln. Sie können Ihnen kontospezifische Hinweise geben. Allgemeine Informationen zu SQS-Richtlinien finden Sie in der Dokumentation zu den AWS SQS-Richtlinien.

12. Wie funktioniert die Jobplanung mit Batch?

    Es ist in Control-M /Stone Branch oder in jeden anderen Distributed Scheduler integriert.

# Umplattformierung von Anwendungen mit Rocket Software (ehemals Micro Focus)

In diesem Handbuch wird der end-to-end Prozess der Neuplattformierung von Mainframe-Anwendungen mithilfe AWS von Mainframe-Modernisierungslösungen unter behandelt. AWS Es beschreibt alle Aufgaben und enthält Informationen zur Konfiguration und zum Betrieb von AWS Mainframe Modernization Runtime auf Amazon, EC2 von der Ersteinrichtung und Analyse bis hin zum Erstellen, Testen und Bereitstellen Ihrer modernisierten Anwendungen auf. AWS Es behandelt auch fortgeschrittene Themen wie die Arbeit mit älteren Datenstrukturen, die Verwendung von Vorlagen und vordefinierten Projekten sowie die Einrichtung der Automatisierung für Streaming-Sitzungen.

Themen


- Rocket Software (ehemals Micro Focus) einrichten (bei Amazon EC2)
- Richten Sie die Automatisierung für Rocket Enterprise Analyzer (ehemals Micro Focus) und Rocket Enterprise Developer Streaming-Sitzungen ein
- Datensätze in Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer) als Tabellen und Spalten anzeigen
- Bearbeiten Sie Datensätze mit den Datendatei-Tools von Rocket Software (ehemals Micro Focus) in Enterprise Developer
- Tutorials für Rocket Software (ehemals Micro Focus)
- Verfügbare Batch-Dienstprogramme in AWS Mainframe Modernization


## Rocket Software (ehemals Micro Focus) einrichten (bei Amazon EC2)

AWS Mainframe Modernization bietet mehrere Amazon Machine Images (AMIs) an, die von Rocket Software (ehemals Micro Focus) lizenzierte Produkte enthalten. Diese AMIs ermöglichen Ihnen die schnelle Bereitstellung von Amazon Elastic Compute Cloud (Amazon EC2) -Instances zur Unterstützung von Rocket Software-Umgebungen, die Sie kontrollieren und verwalten. Dieses Thema enthält die Schritte, die erforderlich sind, um auf diese zuzugreifen und sie zu starten AMIs. Ihre Verwendung AMIs ist völlig optional und sie sind nicht erforderlich, um die Tutorials in diesem Benutzerhandbuch abzuschließen.

Themen


- Voraussetzungen für die Einrichtung von Rocket Software (ehemals Micro Focus) (bei Amazon EC2)
- Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3
- Fordern Sie die Aktualisierung der Zulassungsliste für das Konto an
- Erstellen Sie die AWS Identity and Access Management Rolle
- Gewähren Sie License Manager die erforderlichen Berechtigungen
- Abonnieren Sie die Amazon Machine Images
- Starten Sie eine AWS Mainframe Modernization Instanz von Rocket Software (ehemals Micro Focus)
- Subnetz oder VPC ohne Internetzugang


## Voraussetzungen für die Einrichtung von Rocket Software (ehemals Micro Focus) (bei Amazon EC2)

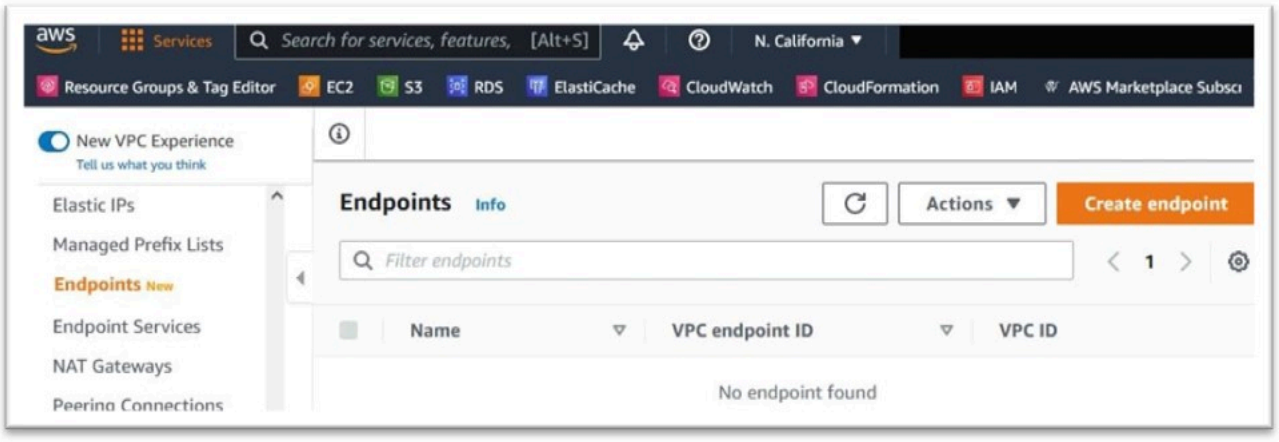
Wenn Sie Rocket Software (bei Amazon EC2) einrichten, stellen Sie sicher, dass Sie die folgenden Voraussetzungen erfüllen.

- Administratorzugriff auf das Konto, auf dem die EC2 Amazon-Instances erstellt werden.
- Identifizieren Sie den AWS-Region Ort, an dem die EC2 Amazon-Instances erstellt werden, und überprüfen Sie, ob der AWS Mainframe Modernization Service verfügbar ist. Weitere Informationen finden Sie unter AWS Services nach Regionen. Stellen Sie sicher, dass Sie eine Region auswählen, in der der Service verfügbar ist.
- Identifizieren Sie die Amazon Virtual Private Cloud (Amazon VPC), in der die EC2 Amazon-Instances erstellt werden.

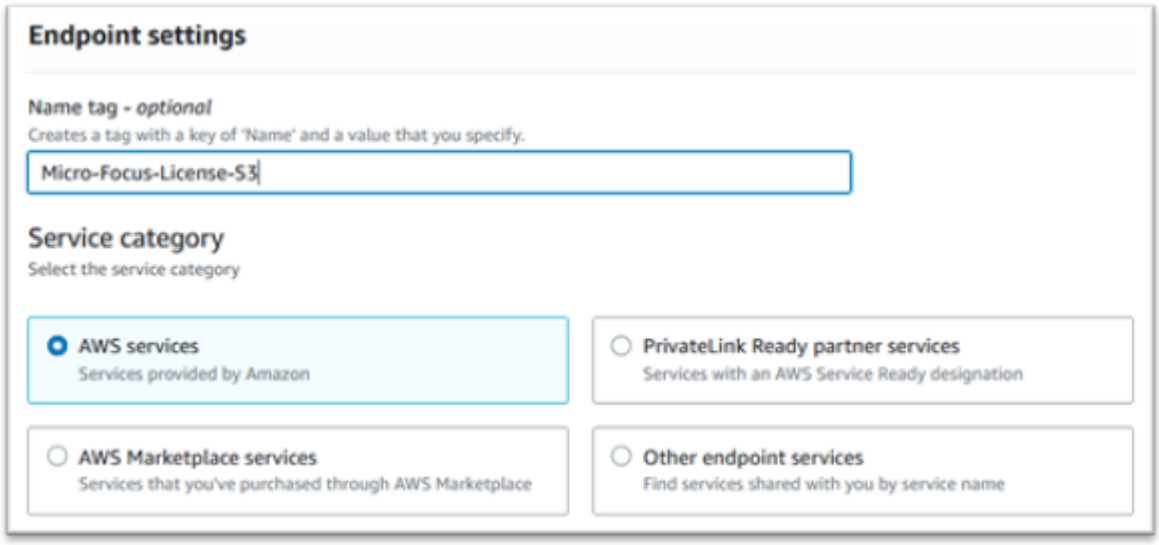
## Erstellen Sie den Amazon VPC-Endpunkt für Amazon S3

In diesem Abschnitt erstellen Sie einen Amazon VPC-Endpunkt, den Amazon S3 verwenden kann. Die Einrichtung dieses Endpunkts hilft Ihnen später bei der Einrichtung des Internetzugangs für VPC.

1. Navigieren Sie zu Amazon VPC in der AWS Management Console.
2. Wählen Sie im Navigationsbereich Endpunkte aus.
3. Wählen Sie Endpunkt erstellen aus.

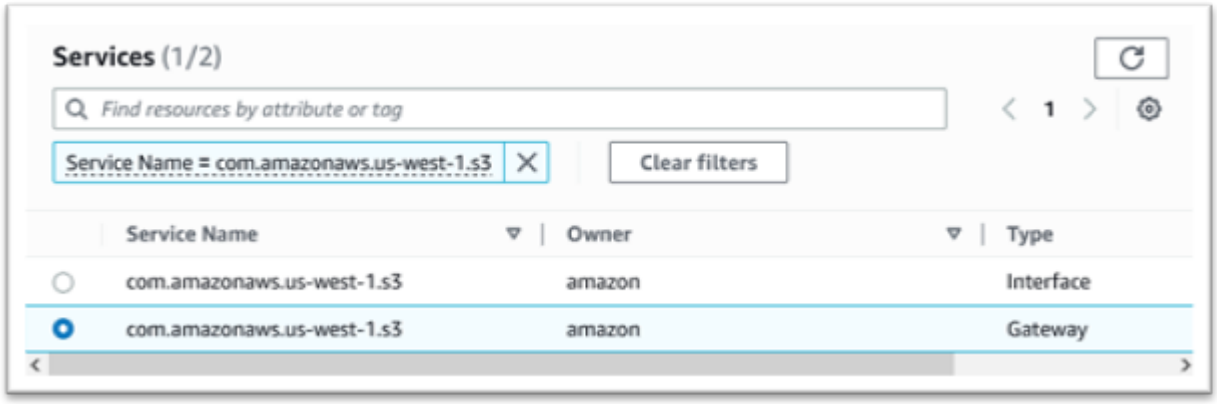



4. Geben Sie ein aussagekräftiges Namensschild ein, zum Beispiel: „Micro-Focus-License-S3“.
5. Wählen Sie AWS-Services als Servicekategorie.

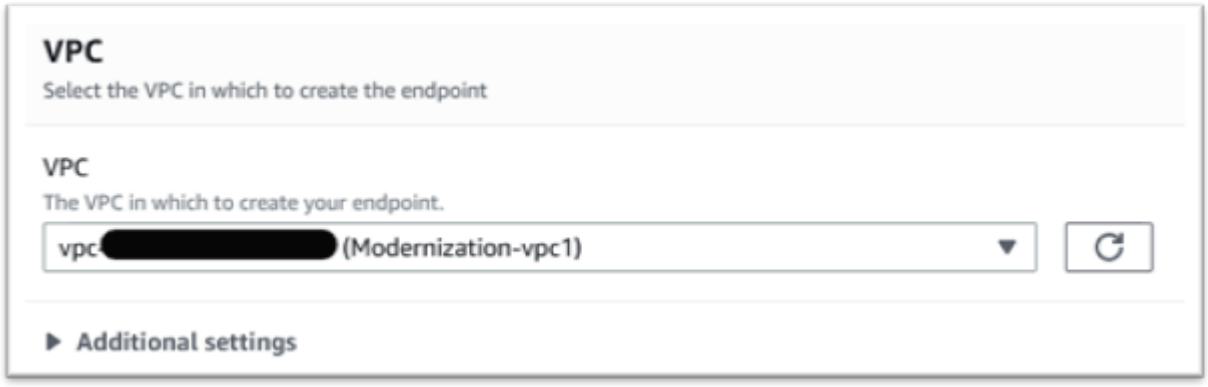



6. Suchen Sie unter Services nach dem Amazon S3 Gateway-Service: com.amazonaws. [Region] .s3.

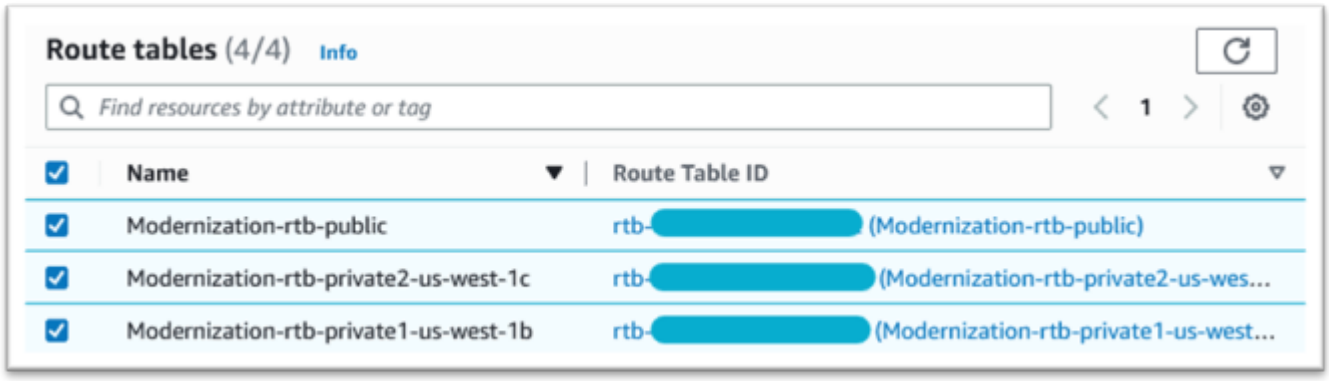
   Denn `us-west-1` das wäre:`com.amazonaws.us-west-1.s3`.
7. Wählen Sie den Gateway-Service.



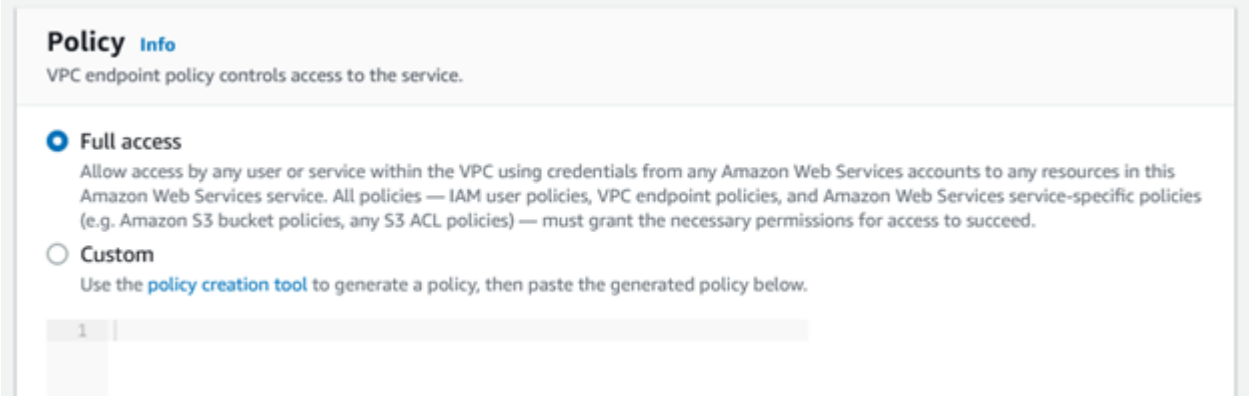

8. Wählen Sie für VPC die VPC aus, die Sie verwenden werden.




9. Wählen Sie alle Routentabellen für die VPC aus.




10. Wählen Sie unter Richtlinie die Option Vollzugriff aus.




> **Note**
>
> Wenn Sie sich entscheiden, eine benutzerdefinierte Richtlinie zu erstellen, stellen Sie sicher, dass diese Zugriff auf den Amazon S3 S3-Bucket hats`3://aws-supernova-marketplace-<region>-prod`.

11. Klicken Sie auf Endpunkt erstellen.

## Fordern Sie die Aktualisierung der Zulassungsliste für das Konto an

Arbeiten Sie mit Ihrem AWS Vertreter zusammen, um Ihr Konto auf die Zulassungsliste für das zu setzen. AWS Mainframe Modernization AMIs 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-xxxxxxxxxxxxxxxxx` ID für die Datei com.amazonaws. [Region] .s3-Gateway-Endpunkt.
- Die Anzahl der Lizenzen, die für alle Rocket Software Enterprise Suite AMI EC2 Amazon-Instances erforderlich sind.

  Eine Lizenz ist pro CPU-Kern erforderlich (pro 2 V CPUs für die meisten EC2 Amazon-Instances).

  Weitere Informationen finden Sie unter CPU-Optionen optimieren.

  Die angeforderte Nummer kann in future von angepasst werden AWS.

> **Note**
>
> Wenden Sie sich an Ihren AWS Vertreter oder an den Mitarbeiter AWS -Support , der in Ihrem Namen das Support-Ticket für die Allowlist-Anfrage öffnet. Es kann nicht direkt von Ihnen angefordert werden und es kann mehrere Tage dauern, bis die Anfrage bearbeitet ist.

## Erstellen Sie die AWS Identity and Access Management Rolle

Erstellen Sie eine AWS Identity and Access Management Richtlinie und eine Rolle, die von den AWS Mainframe Modernization EC2 Amazon-Instances 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-Profils zu den EC2 Amazon-Instances können Rocket-Softwarelizenzen zugewiesen werden. Weitere Informationen zu Instance-Profilen finden Sie unter Verwenden einer IAM-Rolle, um Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden, Berechtigungen zu gewähren.

### Eine IAM-Richtlinie erstellen

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.

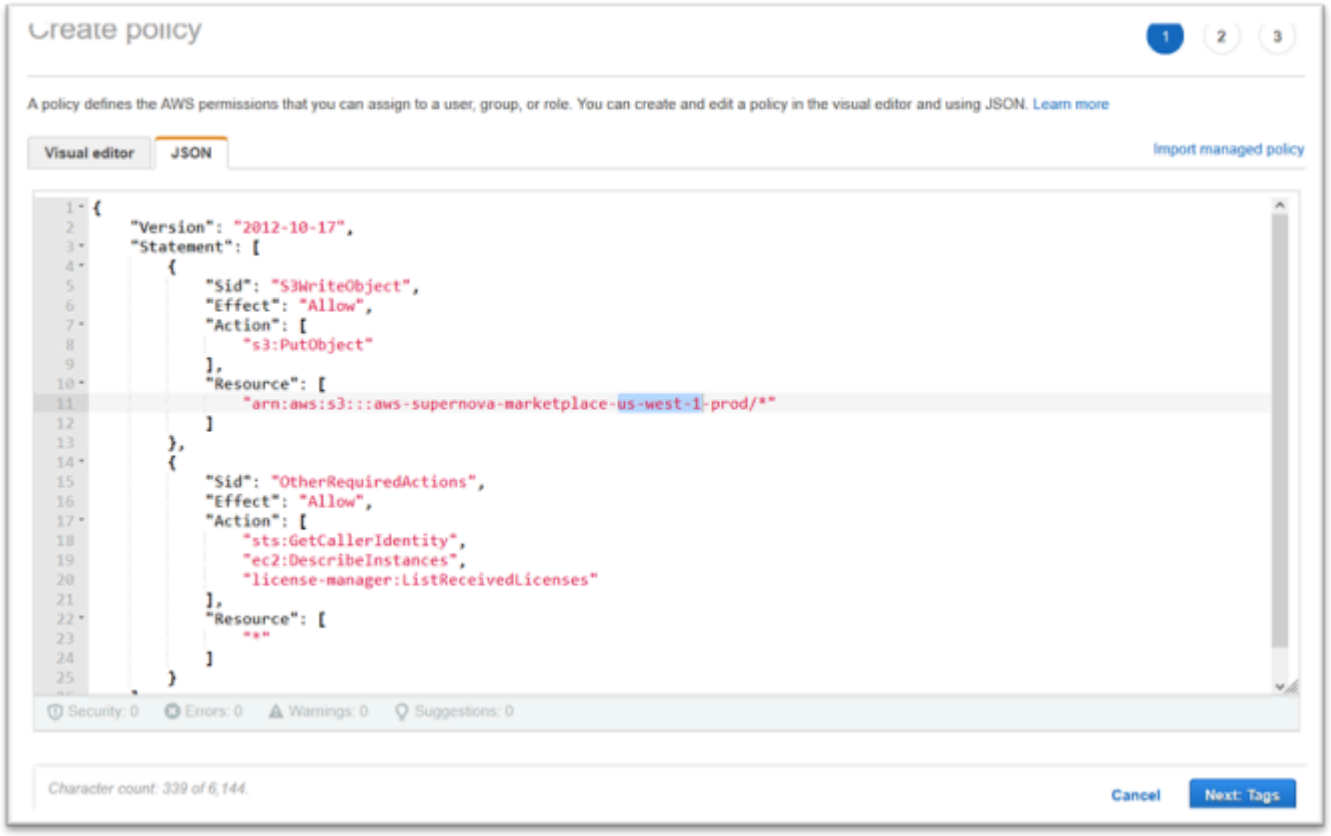



4. Ersetzen Sie `us-west-1` in der folgenden JSON-Datei die AWS-Region Stelle, an der der Amazon S3 S3-Endpunkt definiert wurde, und kopieren Sie dann die JSON-Datei und fügen Sie sie in den Richtlinien-Editor ein.

```
{
    "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": [
                "*"
            ]
        }
    ]
```

```
}
```

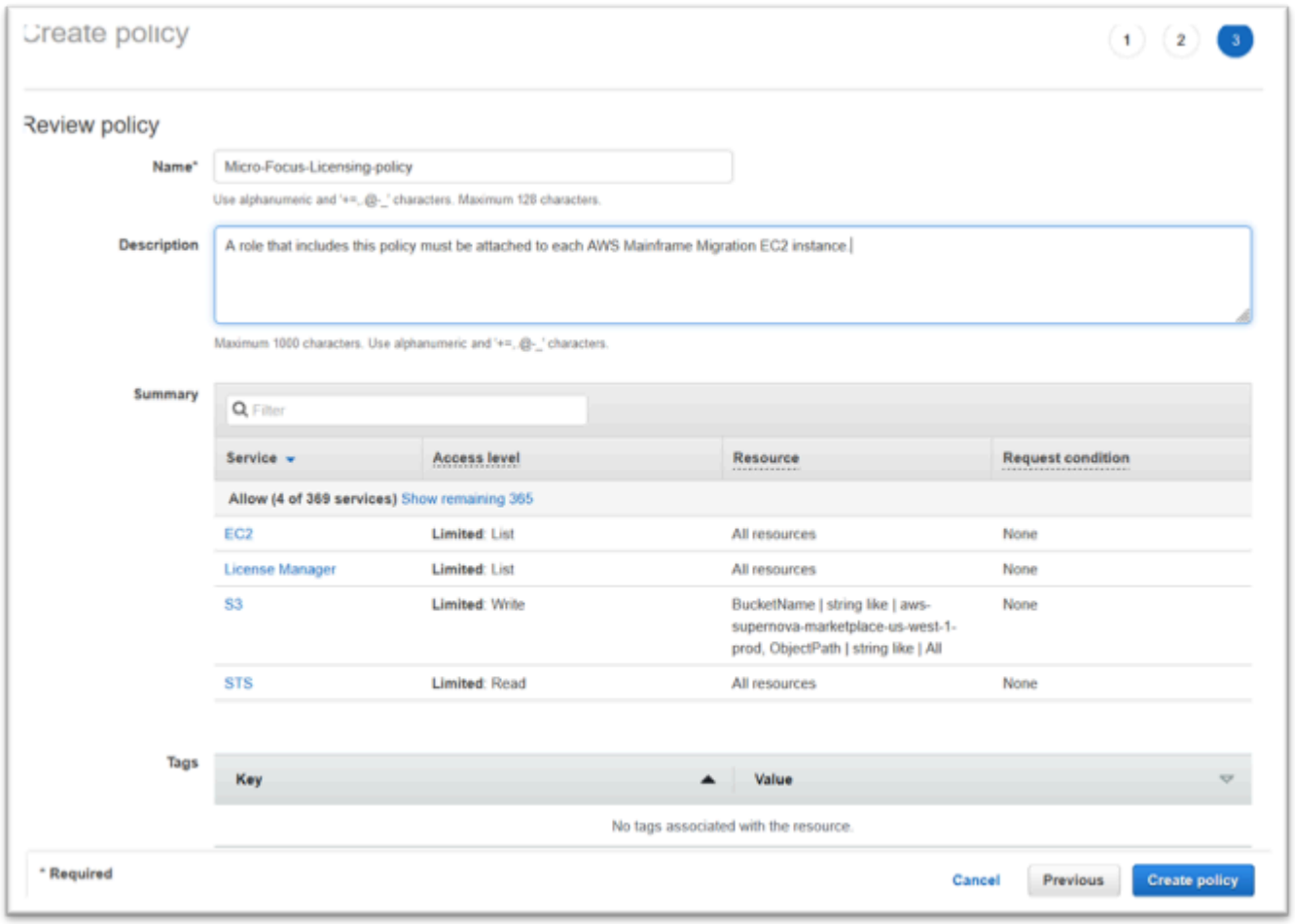
> **Note**
>
> Die Aktionen unter der Sid unterstützen `OtherRequiredActions` keine Berechtigungen auf Ressourcenebene und müssen `*` im Ressourcenelement angegeben werden.




5. Wählen Sie Weiter: Tags aus.




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 beinhaltet, muss jeder AWS Mainframe Modernization EC2 Amazon-Instance zugewiesen werden.“

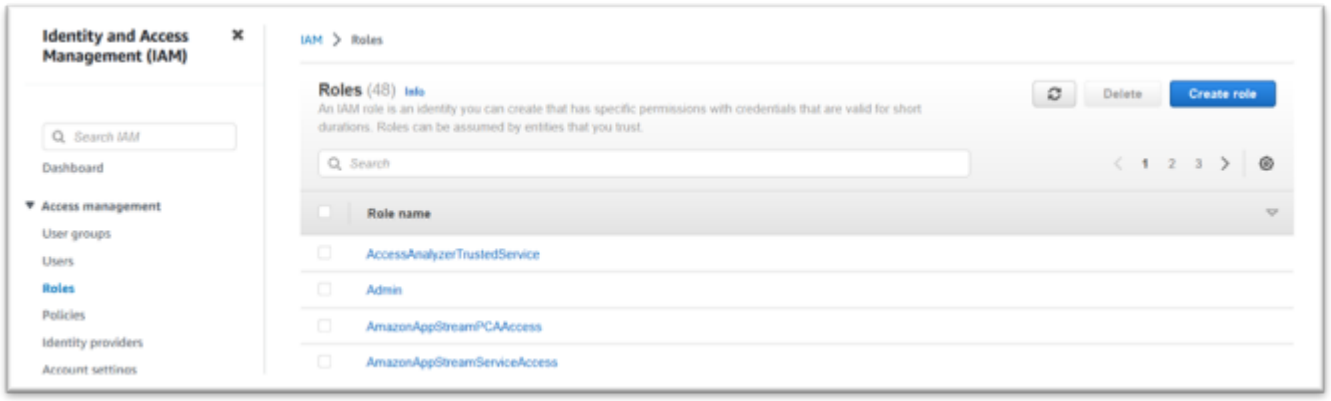


8. Wählen Sie Richtlinie erstellen aus.

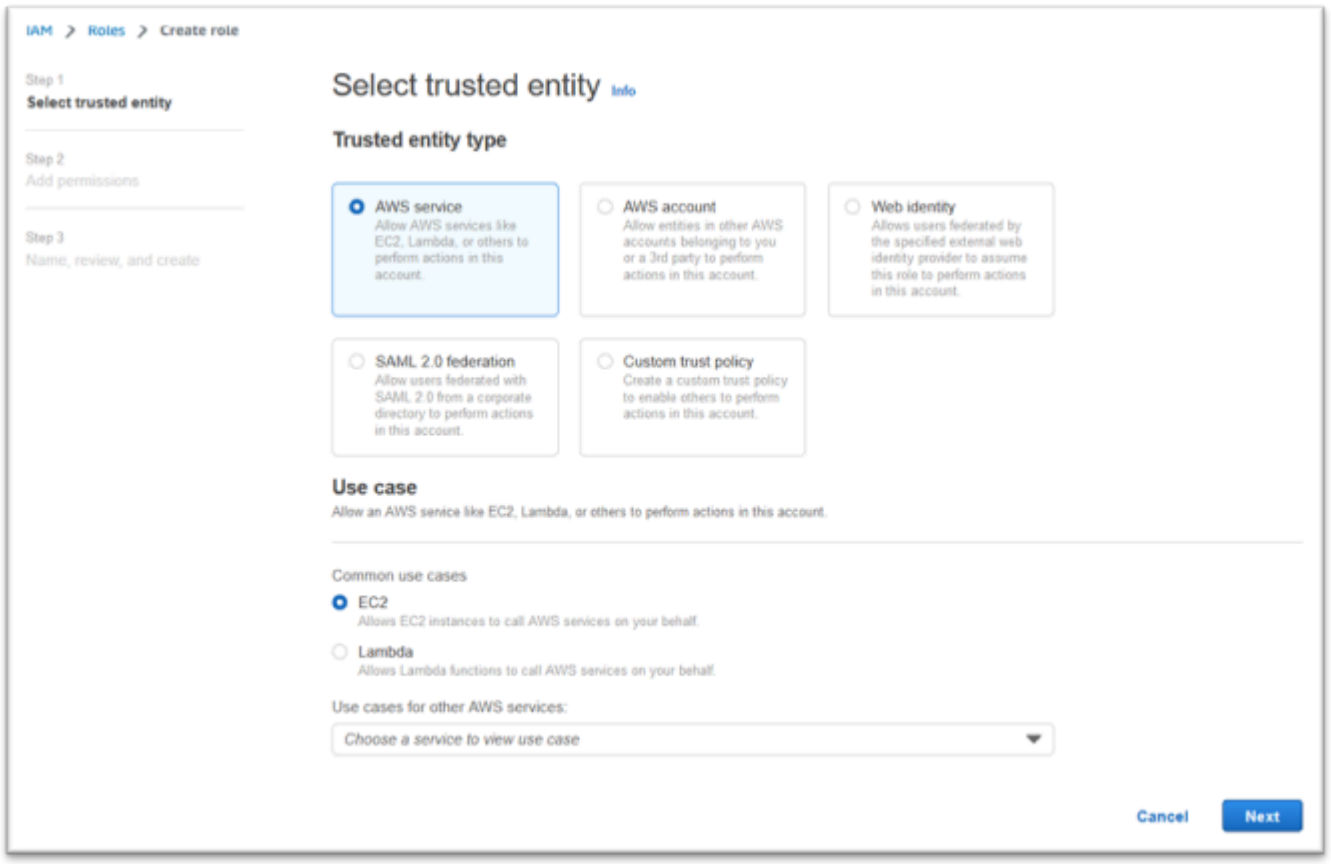
## Erstellen Sie die IAM-Rolle

Nachdem Sie eine IAM-Richtlinie erstellt haben, erstellen Sie eine IAM-Rolle und fügen sie der Richtlinie hinzu.

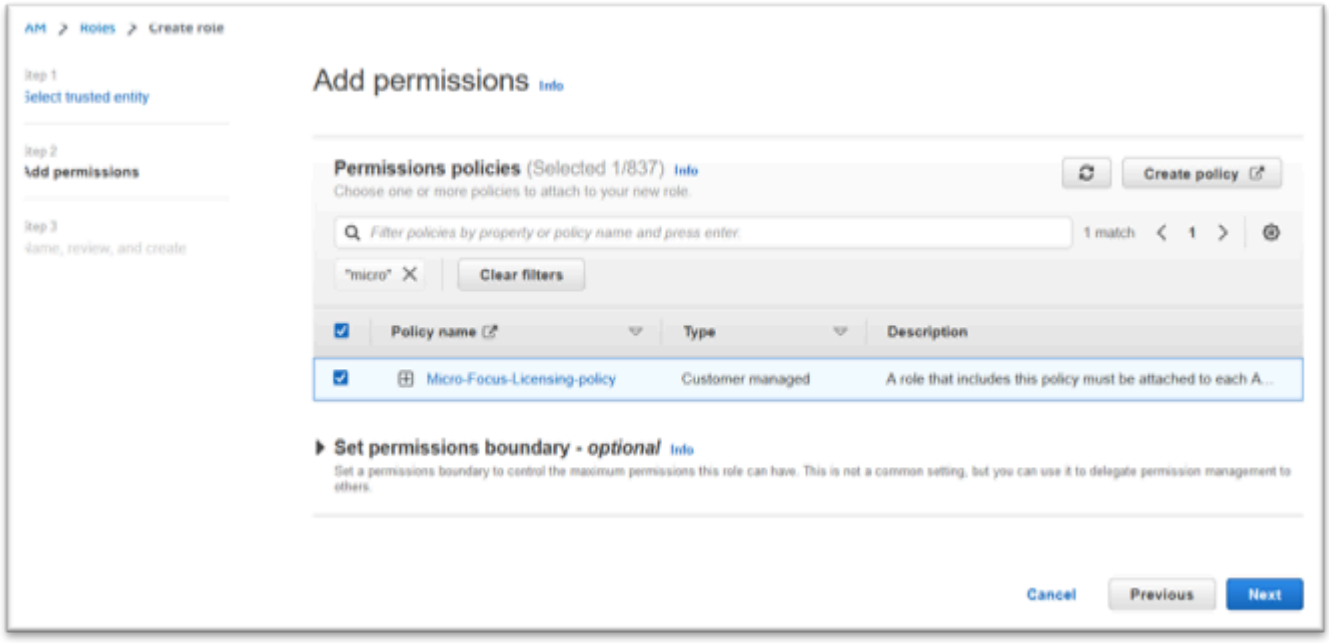
1. Navigieren Sie zu IAM in der. AWS Management Console
2. Wählen Sie Rollen und dann Rolle erstellen aus.



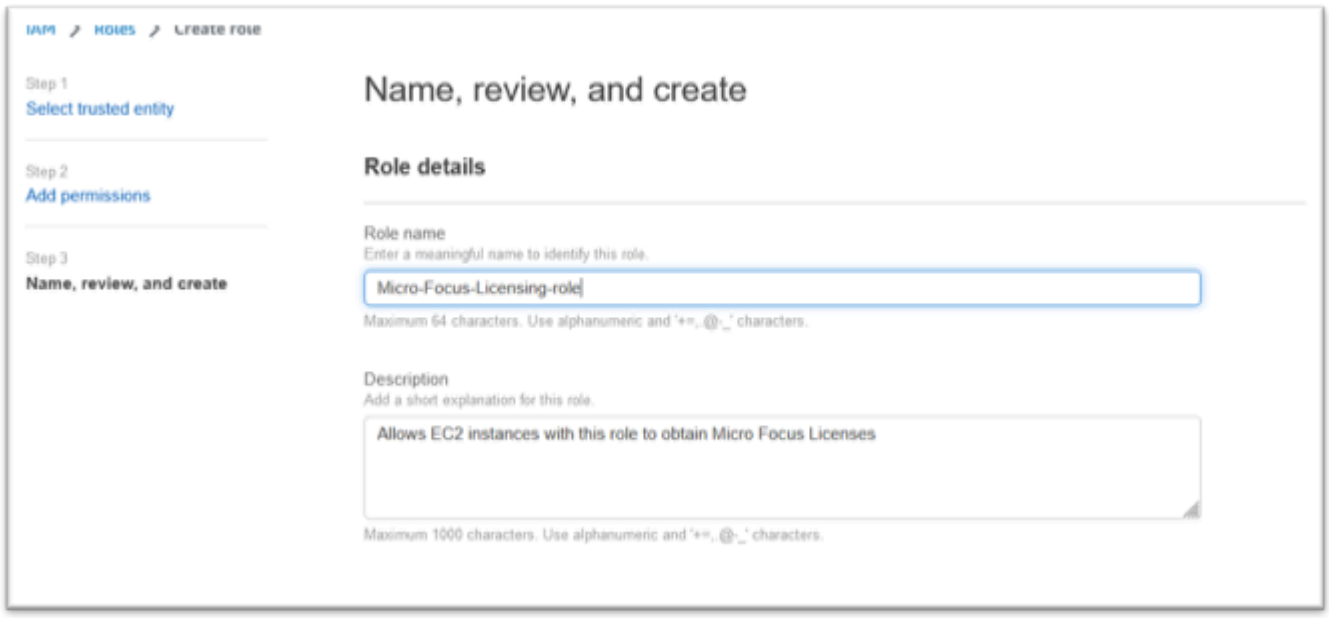

3. Belassen Sie den Entitätstyp Vertrauenswürdig als AWS Dienst und wählen Sie den EC2allgemeinen Anwendungsfall aus.




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 EC2 Amazon-Instances mit dieser Rolle, Micro Focus-Lizenzen zu erhalten“.



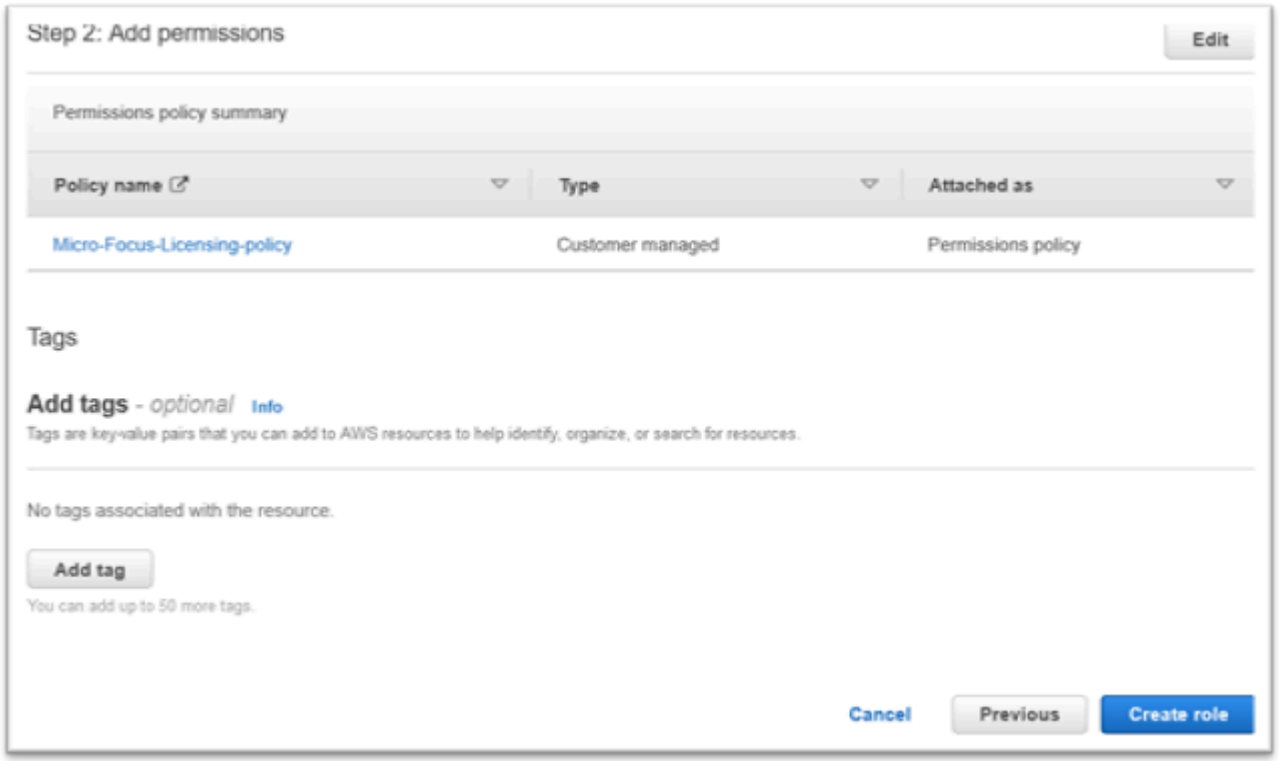

10. Überprüfen Sie unter Schritt 1: Vertrauenswürdige Entitäten auswählen den JSON-Code und stellen Sie sicher, dass er die folgenden Werte hat:

```
{
    "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.



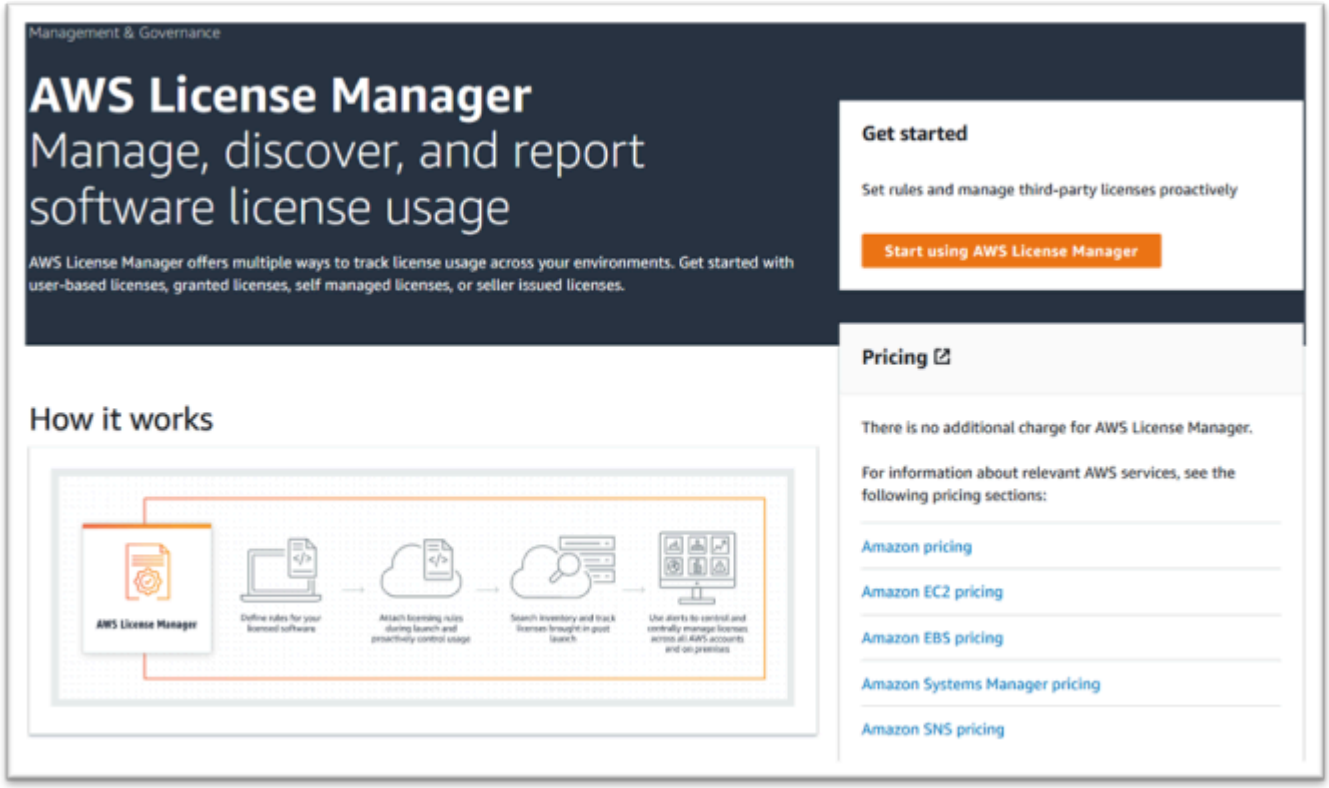

12. Wählen Sie Rolle erstellen aus.

Wenn die Anfrage für die Zulassungsliste abgeschlossen ist, fahren Sie mit den folgenden Schritten fort.

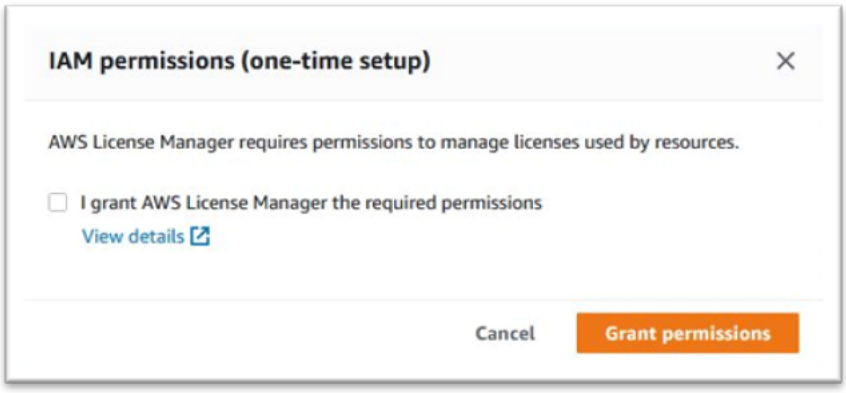
## Gewähren Sie License Manager die erforderlichen Berechtigungen

Sie müssen Ihnen die Erlaubnis erteilen, AWS License Manager die Rocket Software Runtime Engine (bei Amazon EC2) einzurichten.

1. Navigieren Sie zu AWS License Manager in der AWS Management Console.



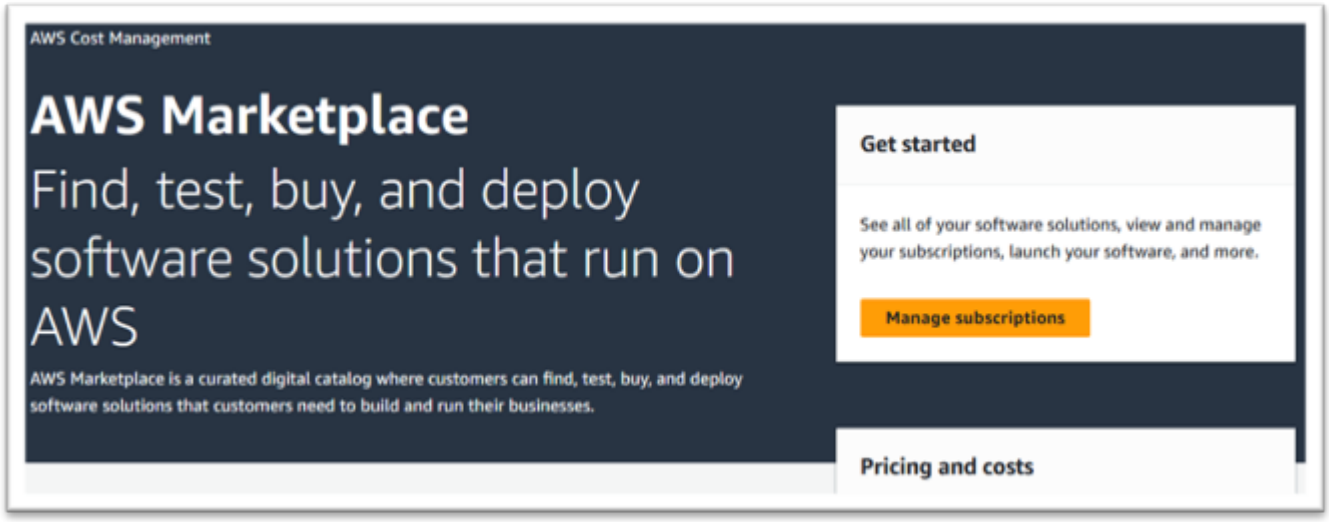

2. Wählen Sie Mit AWS License Manager beginnen aus.
3. Wenn Sie das folgende Pop-up sehen, sehen Sie sich die Details an, aktivieren Sie dann das Kontrollkästchen und klicken Sie auf Berechtigungen gewähren.




## Abonnieren Sie die Amazon Machine Images

Nachdem Sie ein AWS Marketplace Produkt abonniert haben, können Sie eine Instance über das AMI des Produkts starten. Sie können Ihr Abonnement auch verwalten, AMIs wenn Sie die Runtime Engine von Rocket Software (ehemals Micro Focus) (bei Amazon EC2) einrichten.

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**
>
> 1. Wählen Sie nur einen Link für eines der Produkte, zu deren Verwendung Sie autorisiert wurden.

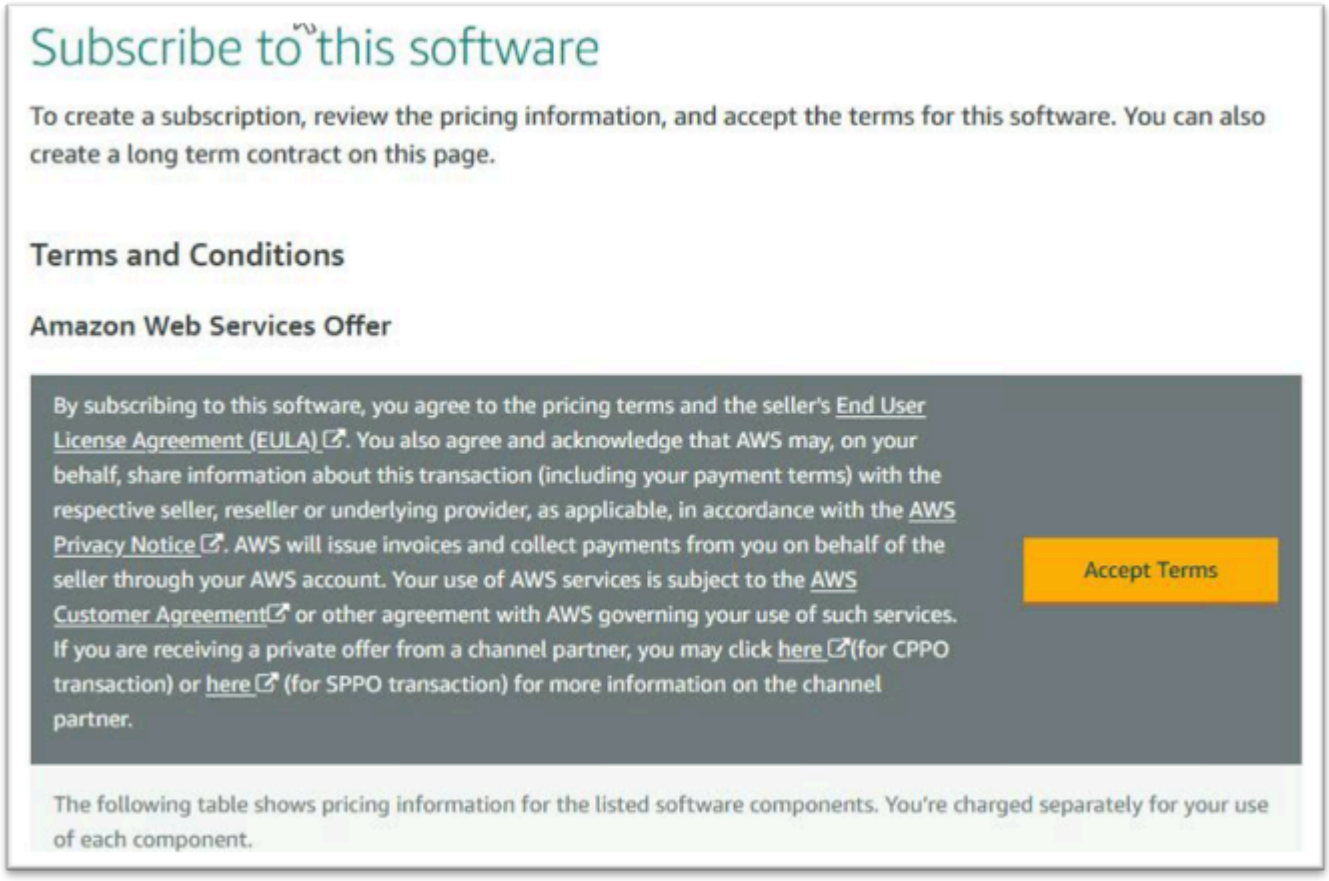
2. Stellen Sie sicher, dass Ihr Konto auf der Zulassungsliste steht, indem Sie der Fordern Sie die Aktualisierung der Zulassungsliste für das Konto an Seite folgen, um diese Links zu verwenden.

- Unternehmensserver: https://aws.amazon.com/marketplace/ pp/prodview-g5emev63l7blc
- Unternehmensserver für Windows: pp/prodview-lwybsiyikbhc2 https://aws.amazon.com/marketplace/
- Unternehmensentwickler: pp/prodview-77qmpr42yzxwk https://aws.amazon.com/marketplace/
- Unternehmensentwickler mit Visual Studio 2022: pp/prodview-m4l3lqiszo6cm https://aws.amazon.com/marketplace/
- Unternehmensanalysator https://aws.amazon.com/marketplace/: pp/prodview-tttheylcmcihm
- Enterprise Build-Tools für Windows: pp/prodview-2rw35bbt6uozi https://aws.amazon.com/marketplace/
- Gespeicherte https://aws.amazon.com/marketplace/Enterprise-Prozeduren: pp/prodview-zoeyqnsdsj6ha
- Gespeicherte Unternehmensprozeduren mit SQL Server 2019: pp/prodview-ynfklquwubnz4 https://aws.amazon.com/marketplace/

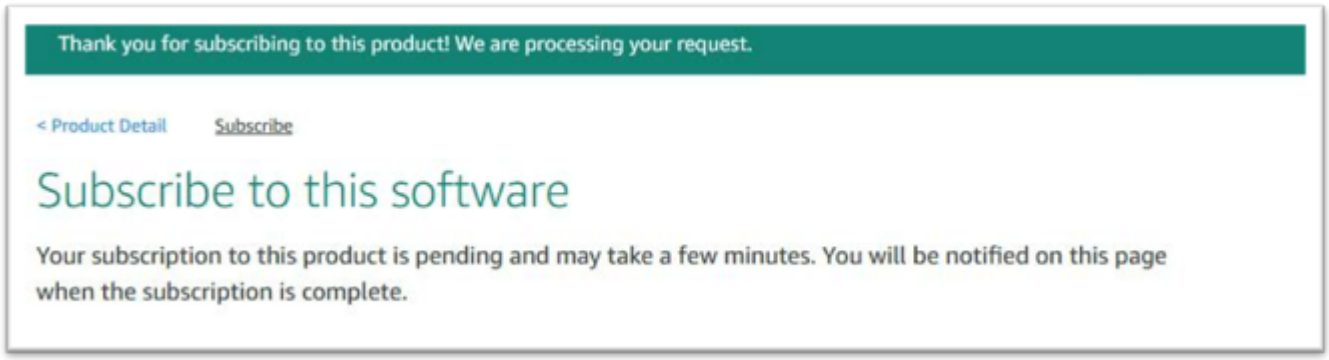
4. Wählen Sie Continue to Subscribe (Weiter abonnieren) aus.



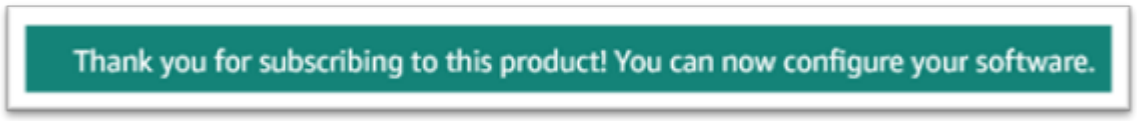

5. Wenn die Allgemeinen Geschäftsbedingungen akzeptabel sind, wählen Sie Bedingungen akzeptieren.




6. Die Bearbeitung des Abonnements kann einige Minuten dauern.

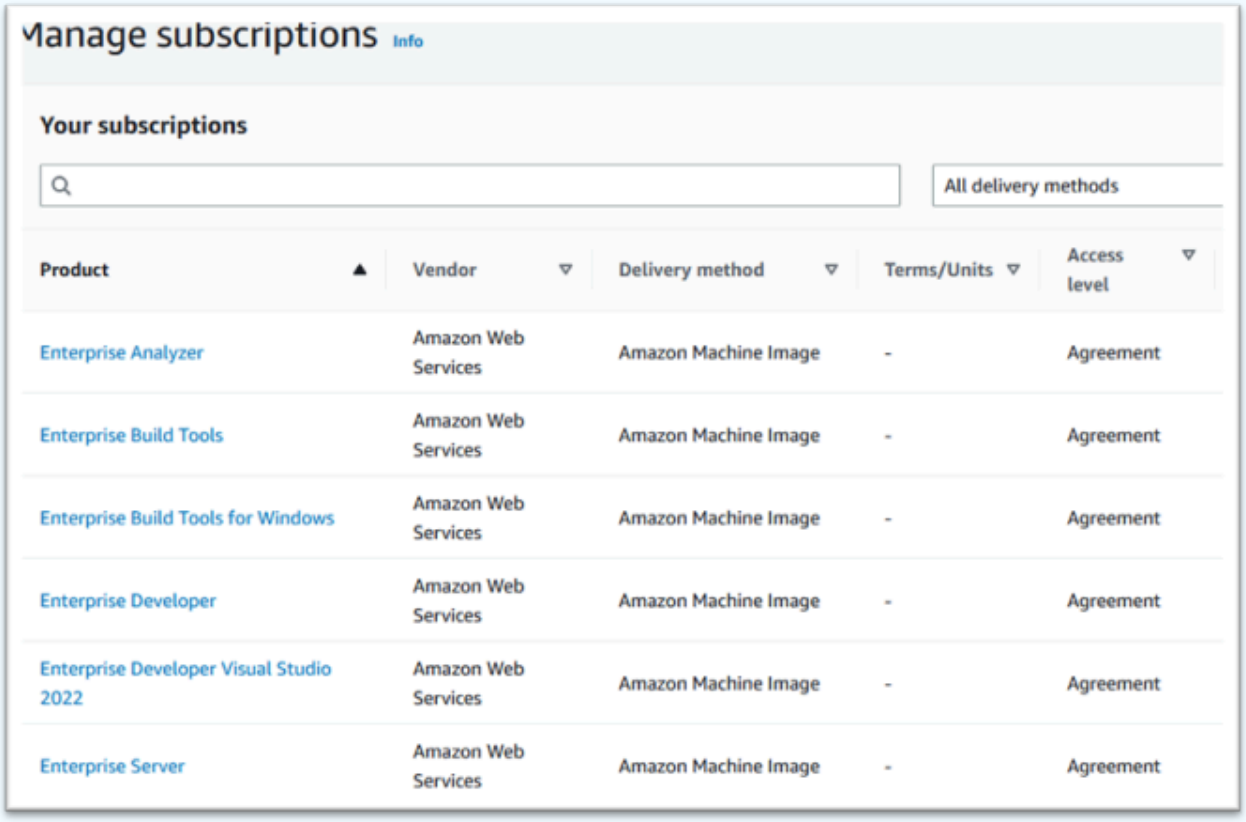



7. Nachdem die Dankesnachricht angezeigt wird, kopieren Sie den nächsten Link aus Schritt 3 und fügen Sie ihn ein, um weitere Abonnements hinzuzufügen.




8. Beenden Sie, wenn unter Abonnements verwalten alle Ihre Abonnenten angezeigt werden. AMIs

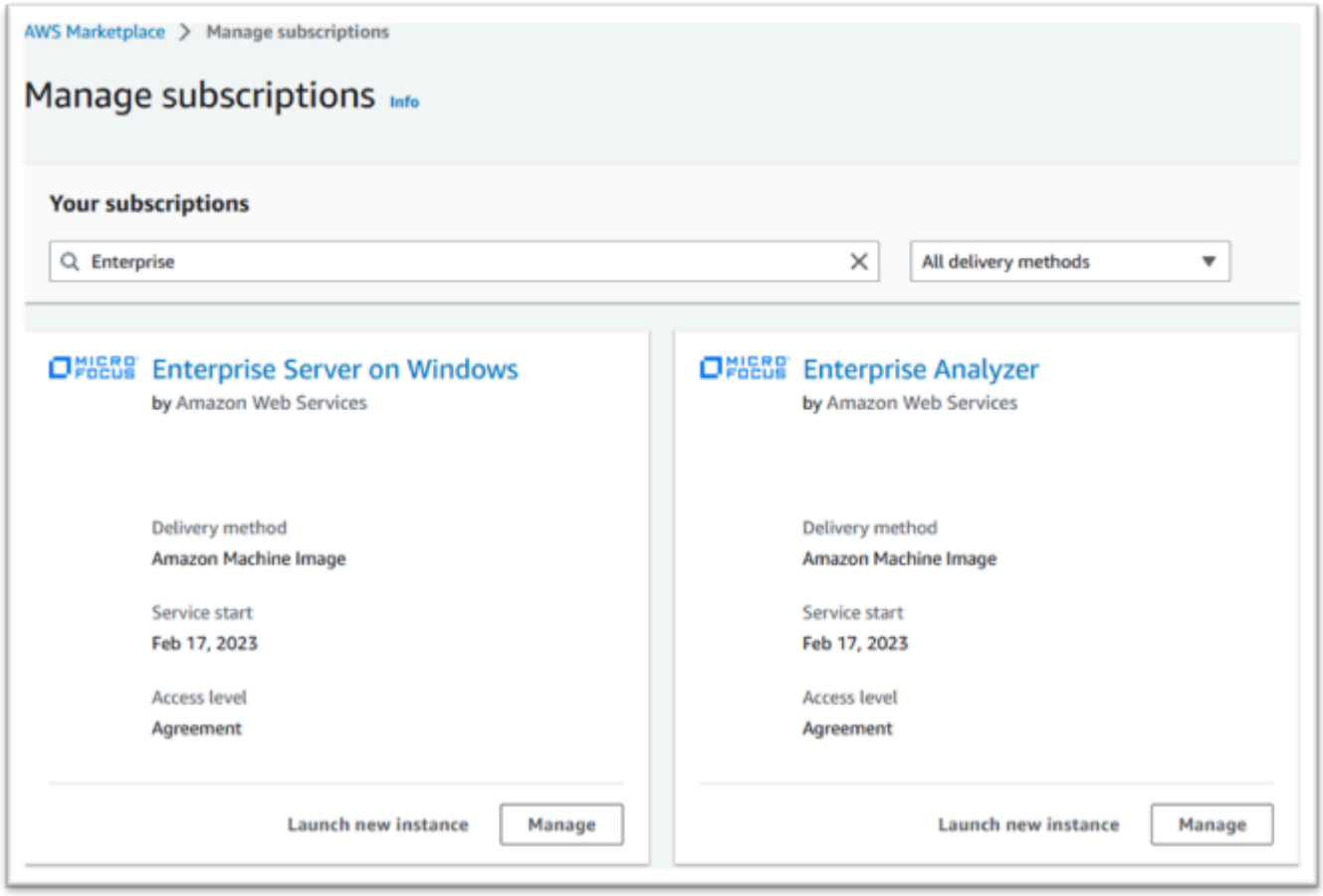
> **Note**
>
> Die Panel-Einstellungen (Zahnradsymbol) sind so eingestellt, dass die Ansicht als Tabelle angezeigt wird.




## Starten Sie eine AWS Mainframe Modernization Instanz von Rocket Software (ehemals Micro Focus)

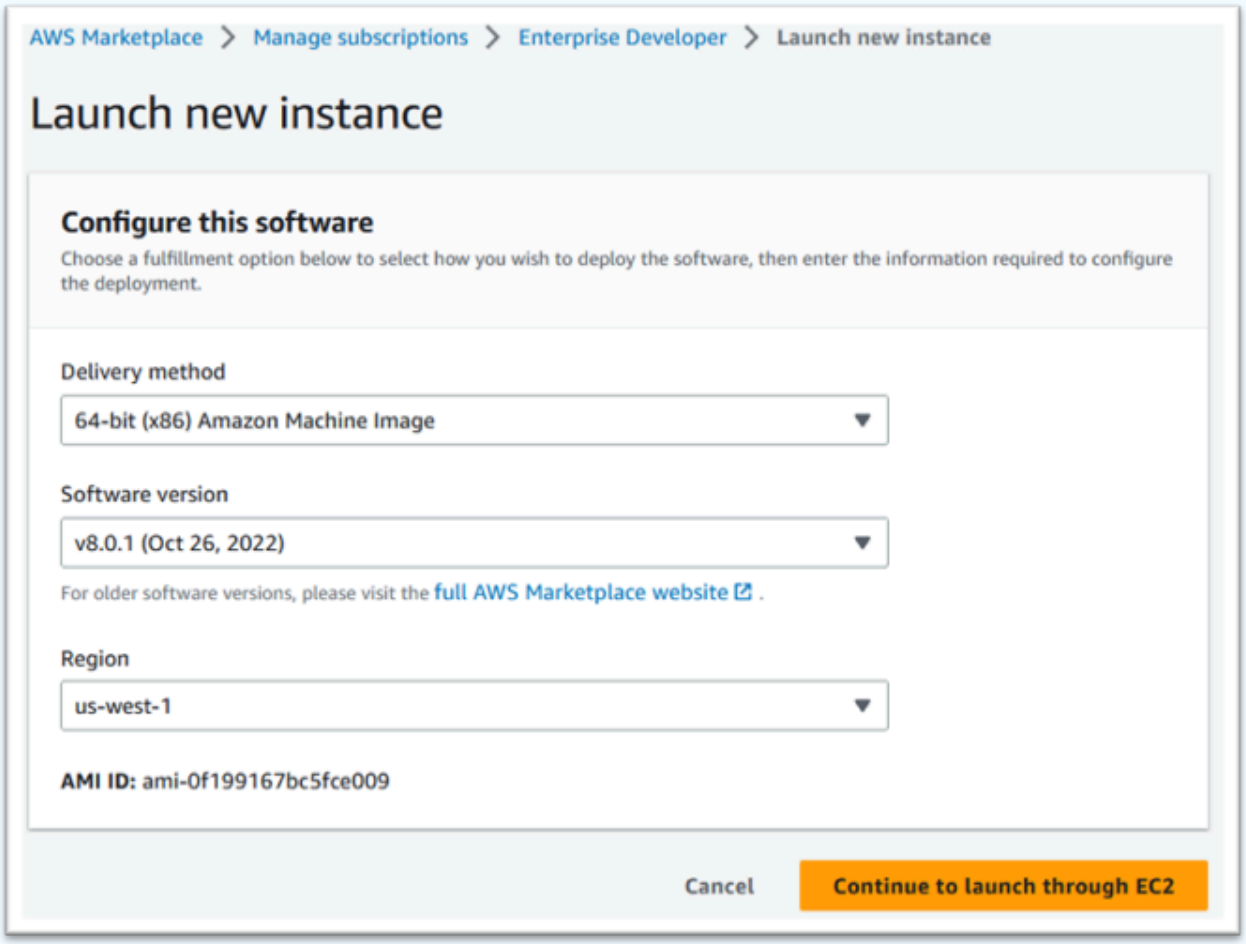
Nachdem Sie die Endpoints, die IAM-Richtlinie und die IAM-Rolle erstellt und das Abonnement abgeschlossen haben, sind Sie bereit AMIs, eine AWS Mainframe Modernization Rocket Software (Micro Focus) -Instanz in der zu starten. AWS Management Console

1. Navigieren Sie zu Abonnements in der. AWS Marketplace AWS Management Console
2. Suchen Sie das AMI, das gestartet werden soll, und wählen Sie Launch New Instance aus.




3. Stellen Sie im Dialogfeld „Neue Instanz starten“ sicher, dass die Region auf der Zulassungsliste ausgewählt ist.
4. Drücken Sie auf Weiter, um den Vorgang zu starten. EC2

> **Note**
>
> Das folgende Beispiel zeigt den Start eines Enterprise Developer AMI, aber der Prozess ist für alle identisch AWS Mainframe Modernization AMIs.

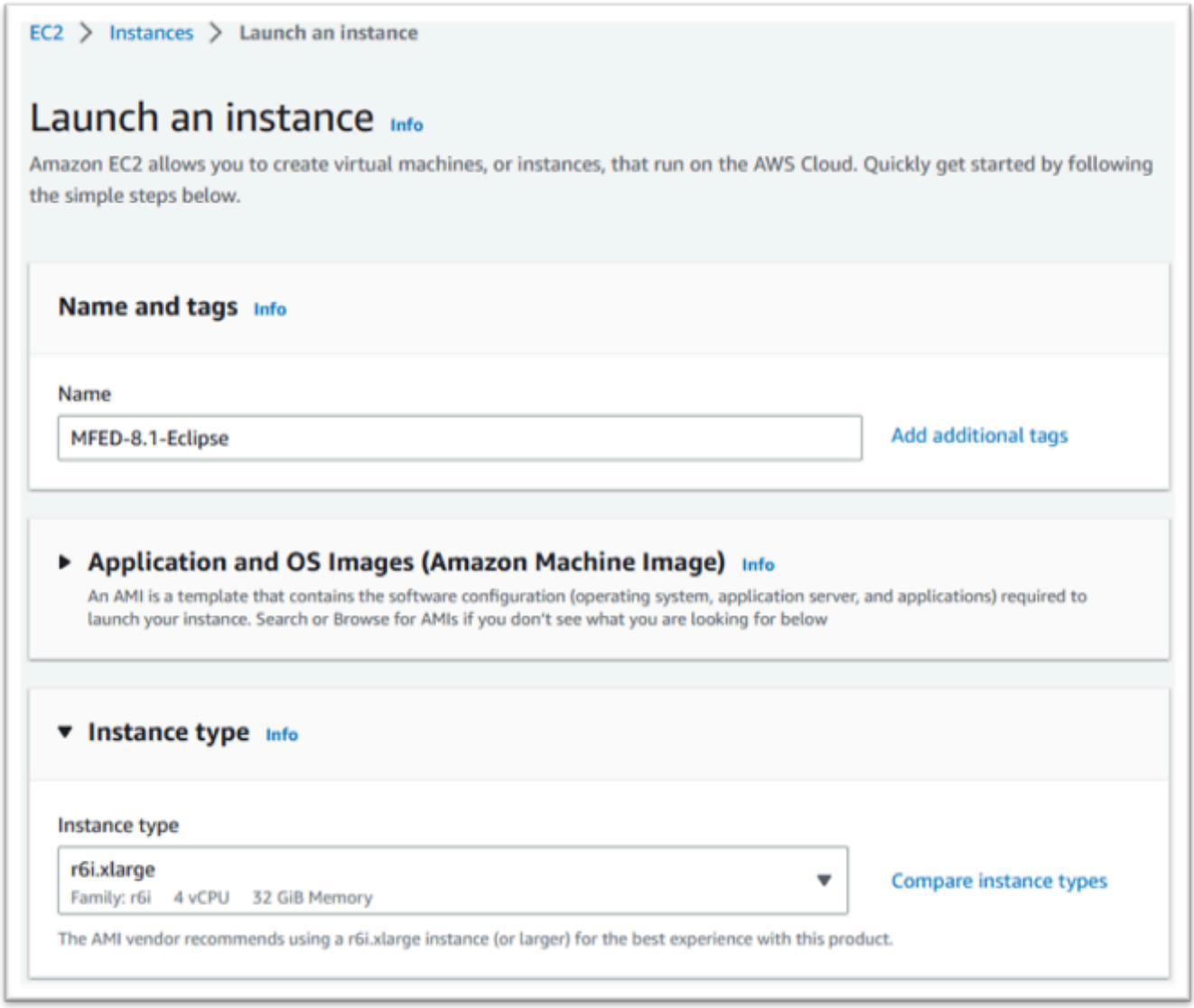



5. Geben Sie einen Namen für den Server ein.
6. Wählen Sie einen Instance-Type.

   Der gewählte Instanztyp sollte sich nach den Leistungs- und Kostenanforderungen des Projekts richten. Es werden folgende 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 Rocket Software Performance Availability Cluster (PAC) mit Scale-Out ein r6i.large

   > **Note**
   >
   > Der Bereich Anwendungs- und Betriebssystem-Images wurde für den Screenshot ausgeblendet.




7. Wählen oder erstellen (und speichern) Sie ein Schlüsselpaar (nicht angezeigt).

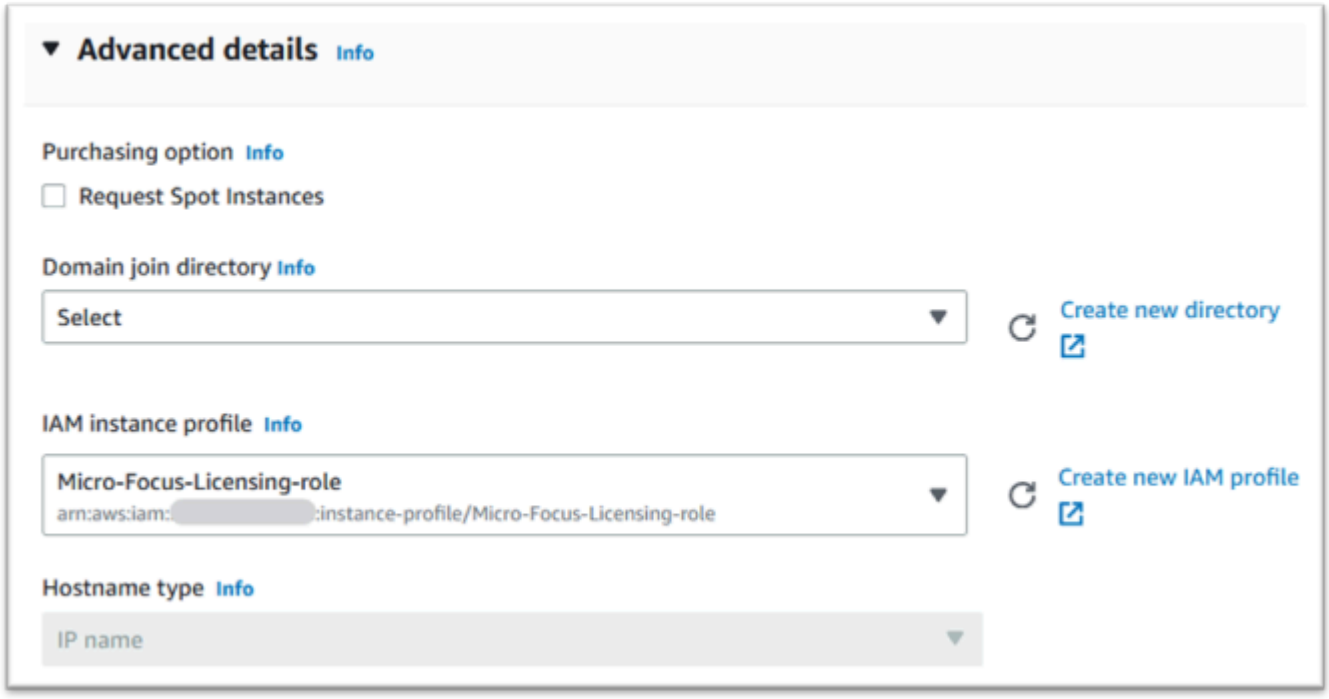
   Weitere Informationen zu Schlüsselpaaren für Linux-Instances finden Sie unter EC2 Amazon-Schlüsselpaare und Linux-Instances.

   Weitere Informationen zu Schlüsselpaaren für Windows-Instances finden Sie unter EC2 Amazon-Schlüsselpaare und Windows-Instances.

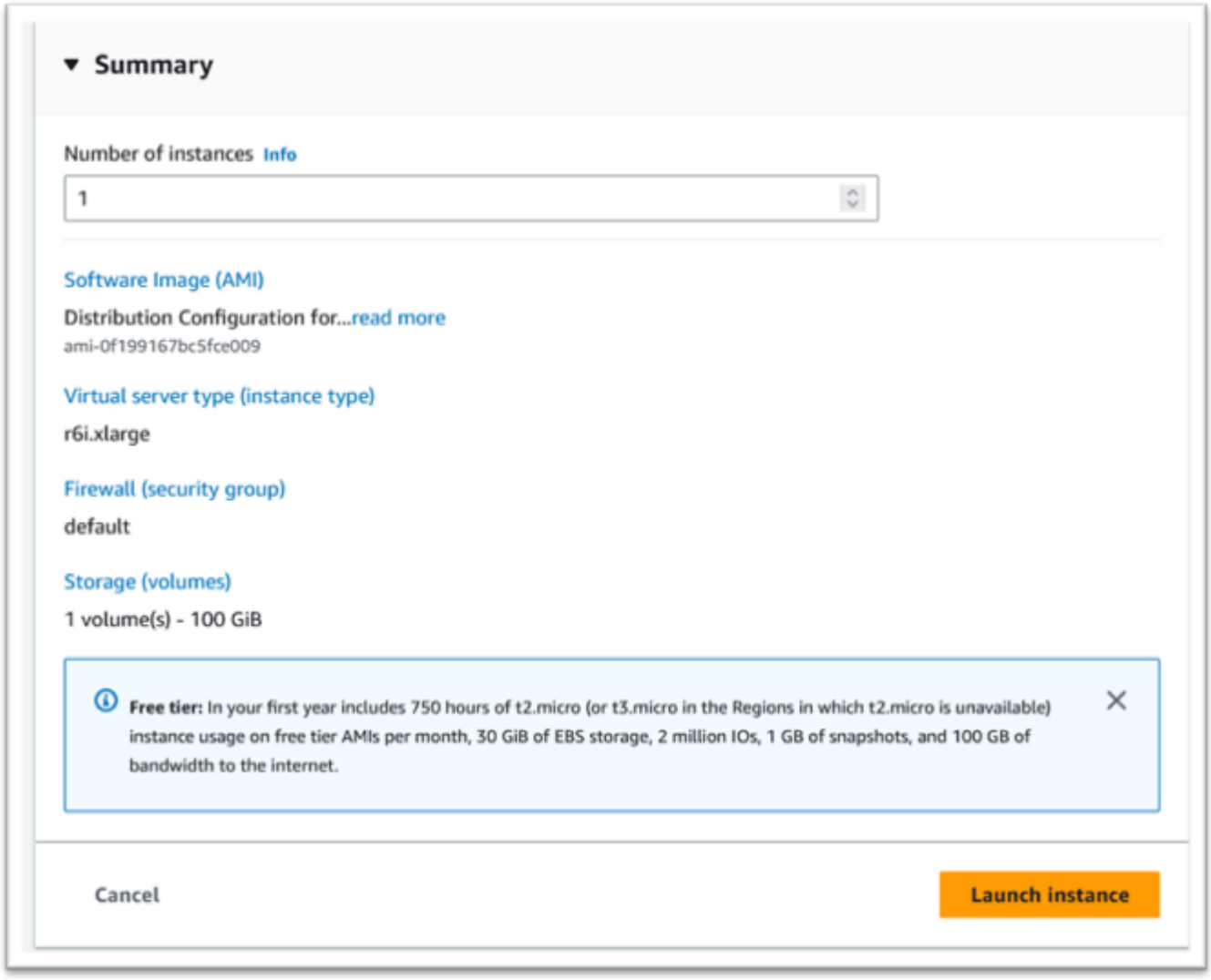
8. Bearbeiten Sie die Netzwerkeinstellungen und wählen Sie die zugelassene VPC und das entsprechende Subnetz aus.

9. Wählen oder erstellen Sie eine Sicherheitsgruppe. Wenn es sich um eine Enterprise EC2 Server-Instanz handelt, ist es üblich, TCP-Verkehr zu den Ports 86 und 10086 zuzulassen, um die Rocket-Software-Konfiguration zu verwalten.
10. Konfigurieren Sie optional den Speicher für die EC2 Amazon-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 Instanz die IAM-Rolle über die Option Sicherheit des Aktionsmenüs für die Instanz ändern. EC2

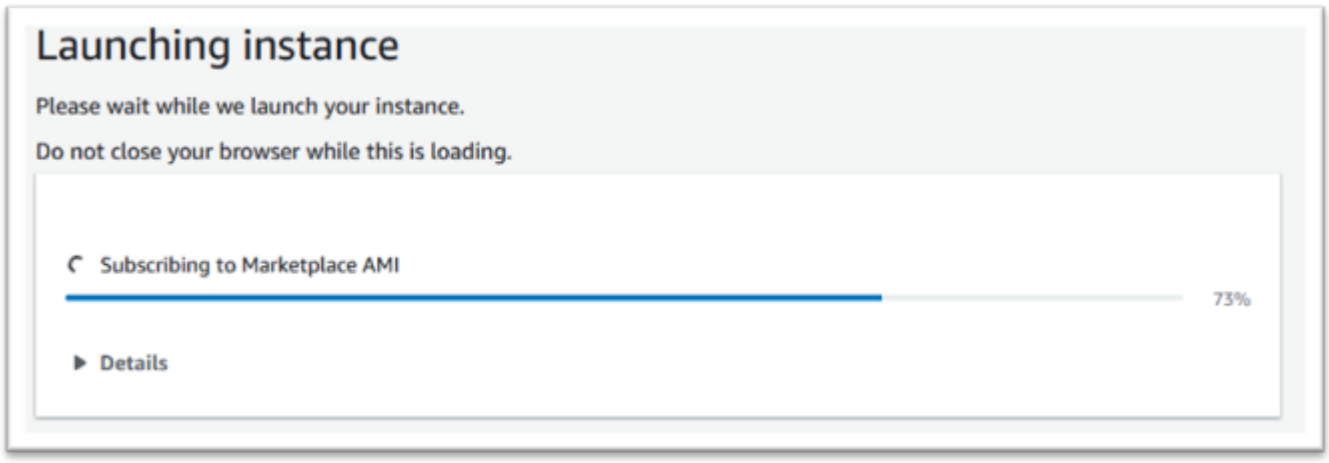



12. Sehen Sie sich die Zusammenfassung an und starten Sie die Instance per Push.

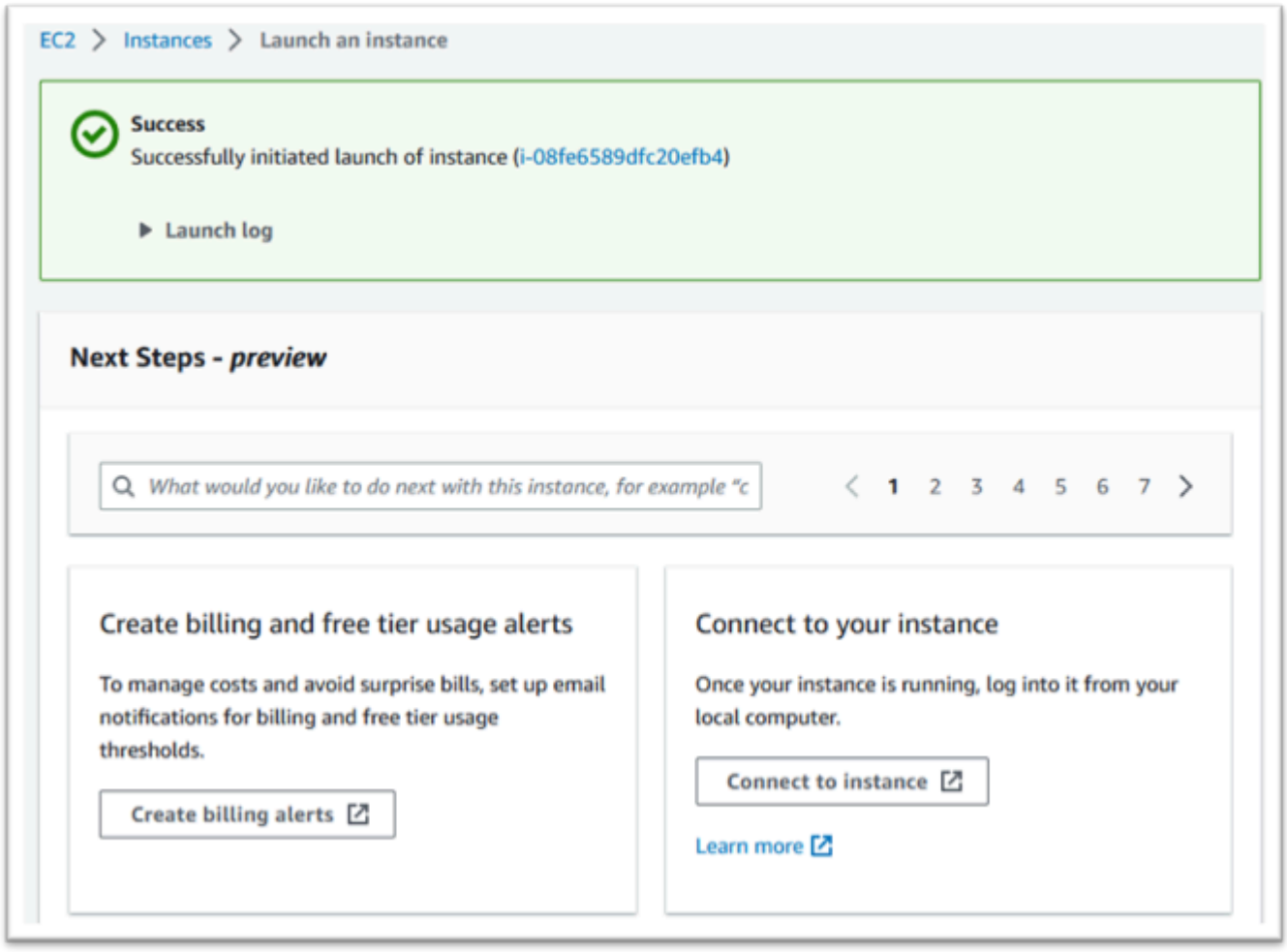



13. Der Instanzstart schlägt fehl, wenn ein ungültiger virtueller Servertyp ausgewählt wird.

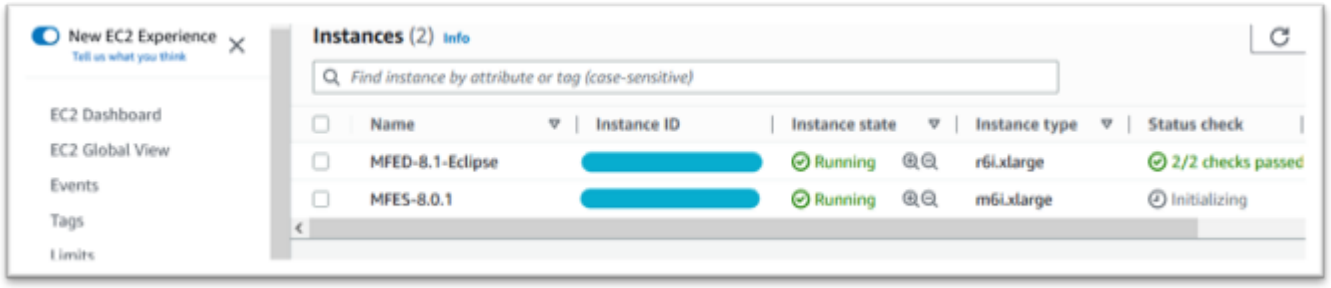
    Wählen Sie in diesem Fall die Option Instanzkonfiguration bearbeiten und ändern Sie den Instanztyp.




14. Sobald die Erfolgsmeldung angezeigt wird, wählen Sie Mit Instanz verbinden, um Verbindungsdetails zu erhalten.




15. Navigieren Sie alternativ zu EC2in der AWS Management Console.
16. Wählen Sie Instances, um den Status der neuen Instanz 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*. 3
- com.amazonaws. *region*.ec2
- com.amazonaws. *region*.lizenzmanager
- 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.

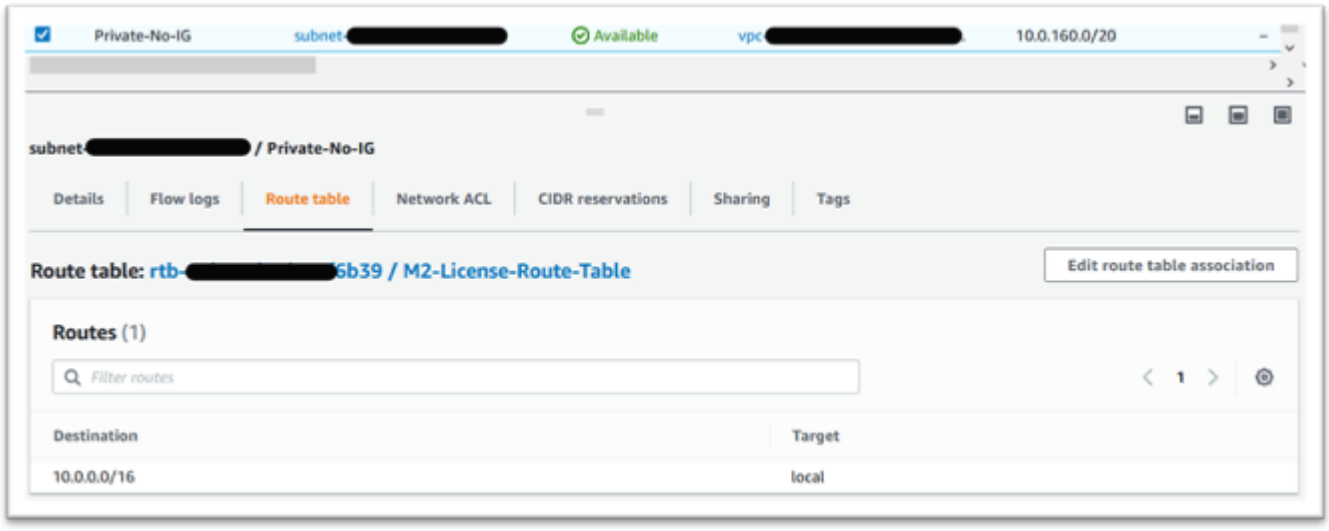
Themen

- Fügen Sie den Routen-Tabelleneintrag für den Amazon S3 S3-Endpunkt hinzu
- Definieren Sie die erforderliche Sicherheitsgruppe
- Erstellen Sie die Dienstendpunkte

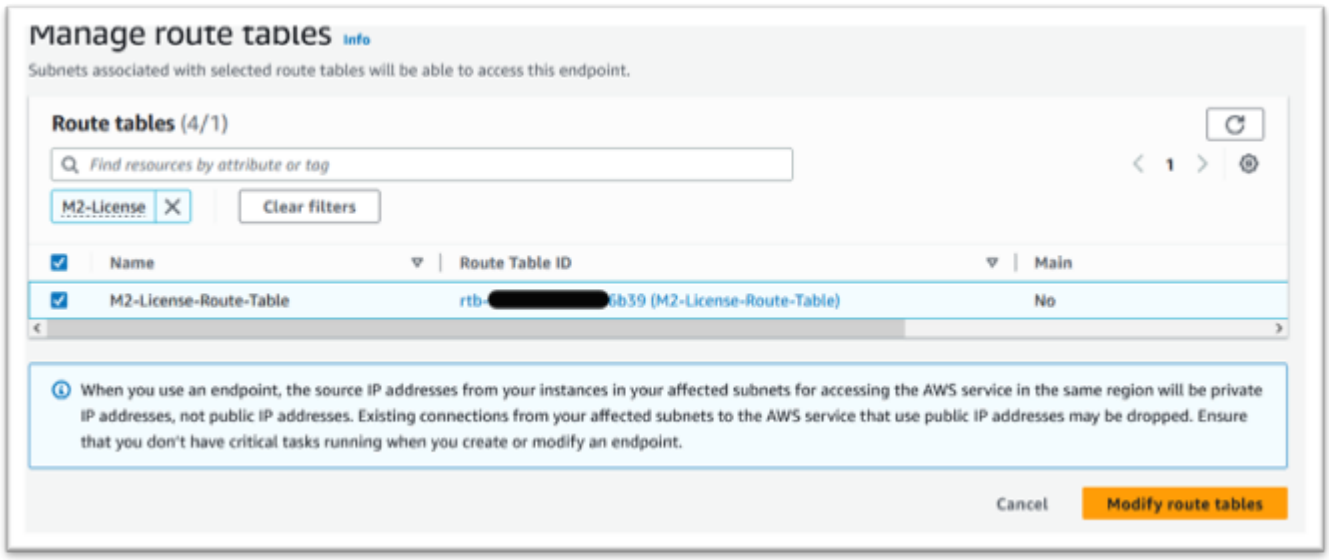
### Fügen Sie den Routen-Tabelleneintrag für den Amazon S3 S3-Endpunkt hinzu

1. Navigieren Sie zu VPC in AWS Management Console und wählen Sie Subnetze aus.
2. Wählen Sie das Subnetz aus, in dem die EC2 Amazon-Instances erstellt werden sollen, und wählen Sie den Tab Route Table.

3. Notieren Sie sich einige nachstehende Ziffern der Routing-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 mit den zuvor identifizierten Ziffern aus und klicken Sie auf Routentabellen ändern.

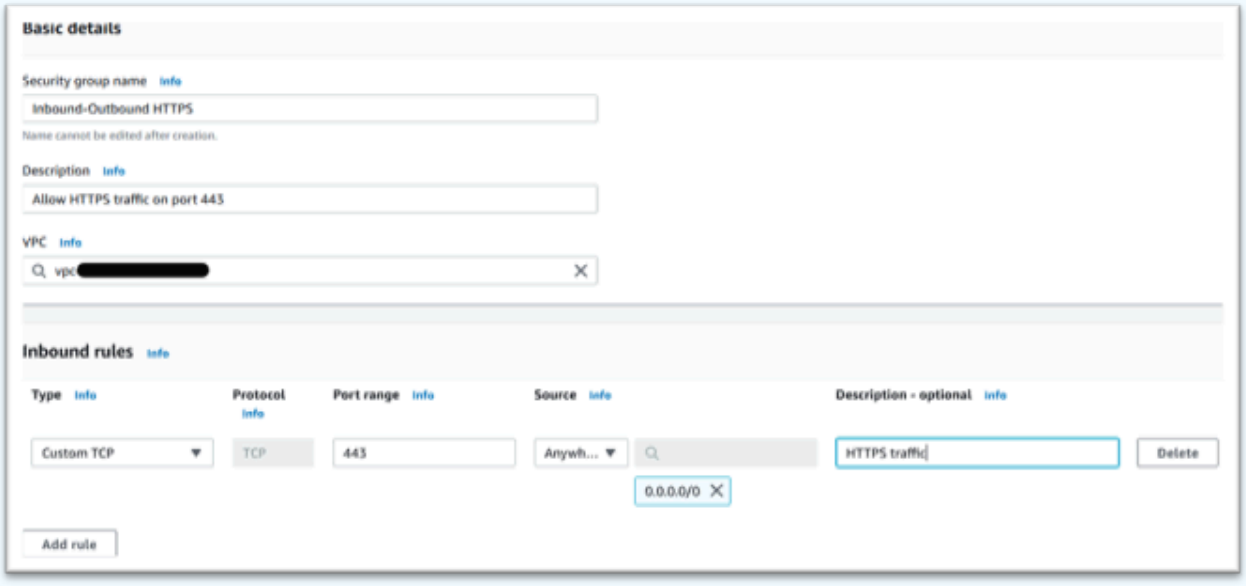


## Definieren Sie die erforderliche Sicherheitsgruppe

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.

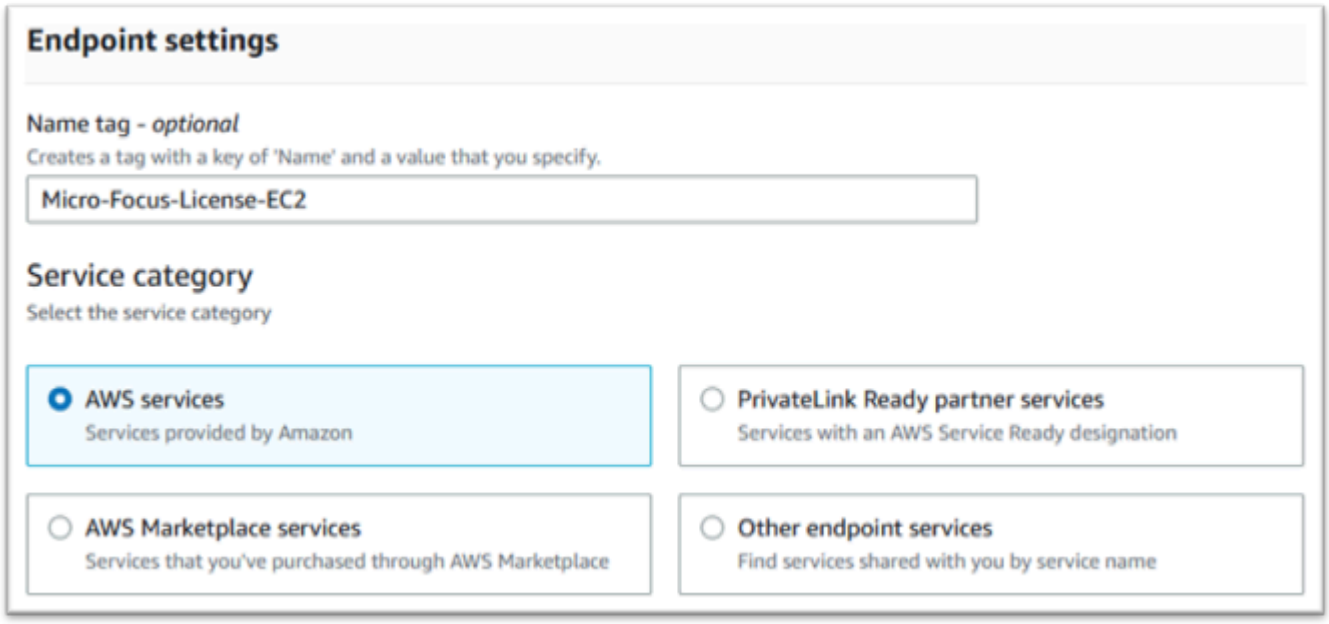



6. Drücken Sie auf Sicherheitsgruppe erstellen.

## 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.



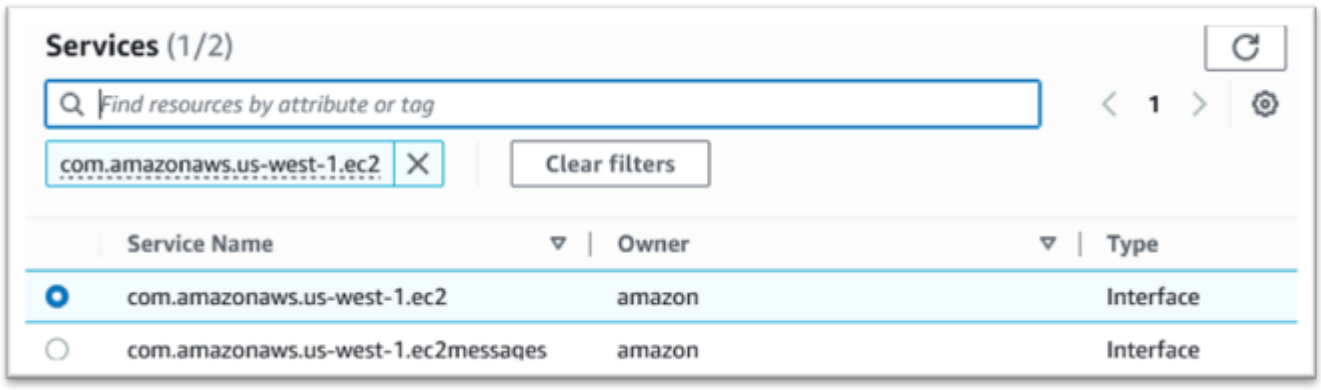

5. Suchen Sie unter Dienste nach dem passenden Interface-Dienst, bei dem es sich um einen der folgenden handelt:

   - „com.amazonaws. *region*.ec2“
   - „com.amazonaws. *region*.sts“
   - „com.amazonaws. *region*.lizenzmanager“

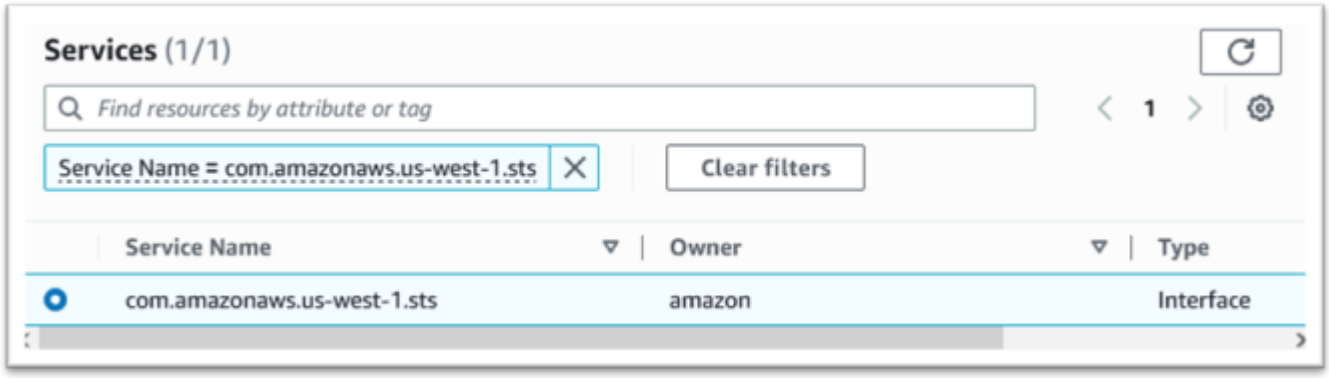
   Zum Beispiel:

   - „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.

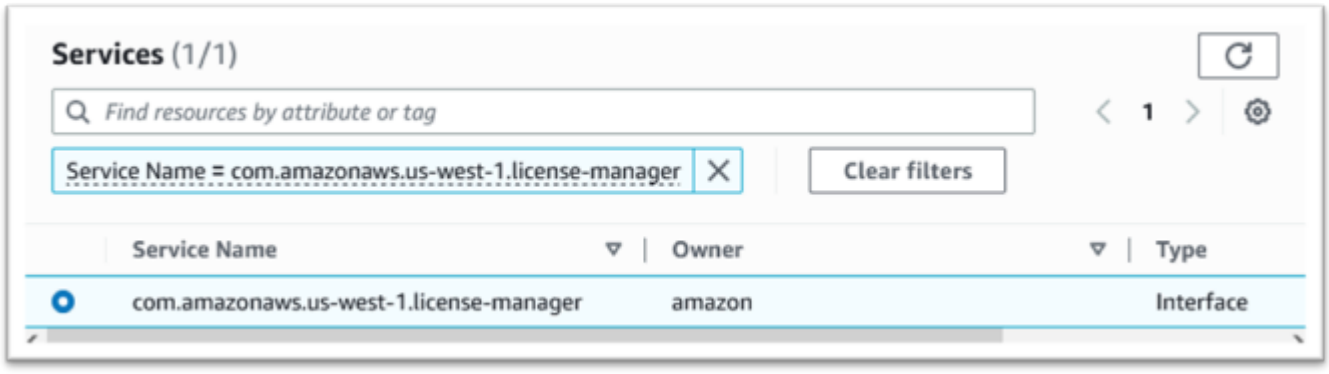
com.amazonaws. ***region***.ec2:



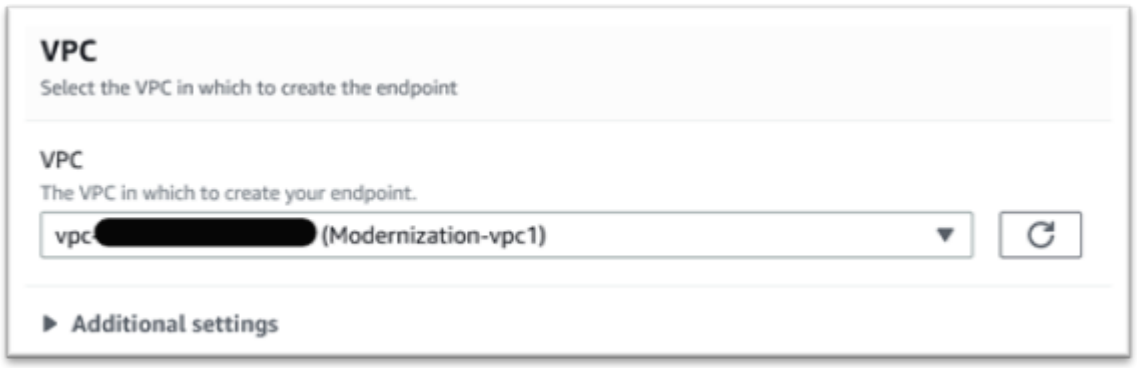

com.amazonaws. ***region***.sts:



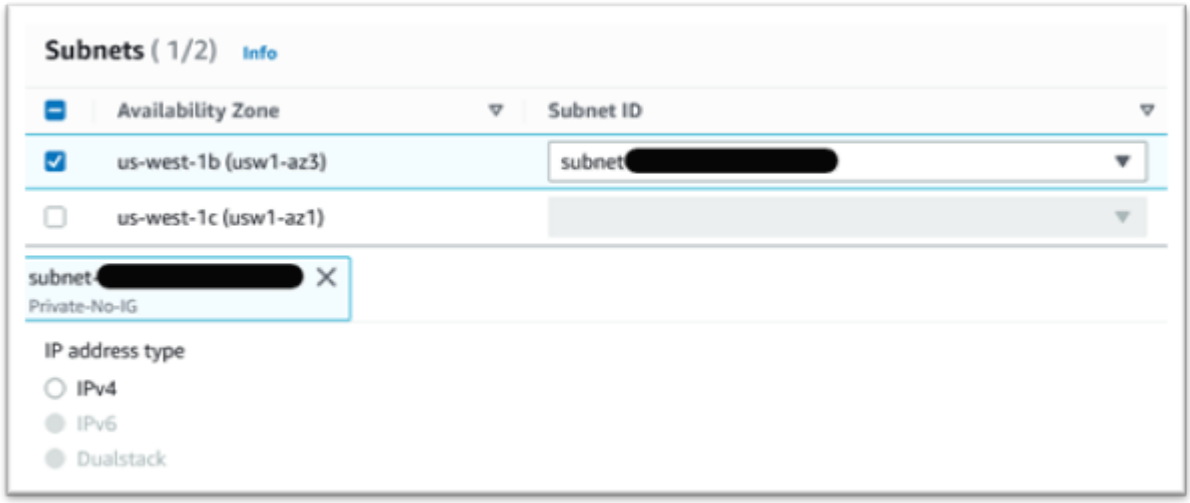

com.amazonaws. ***region***.lizenzmanager:



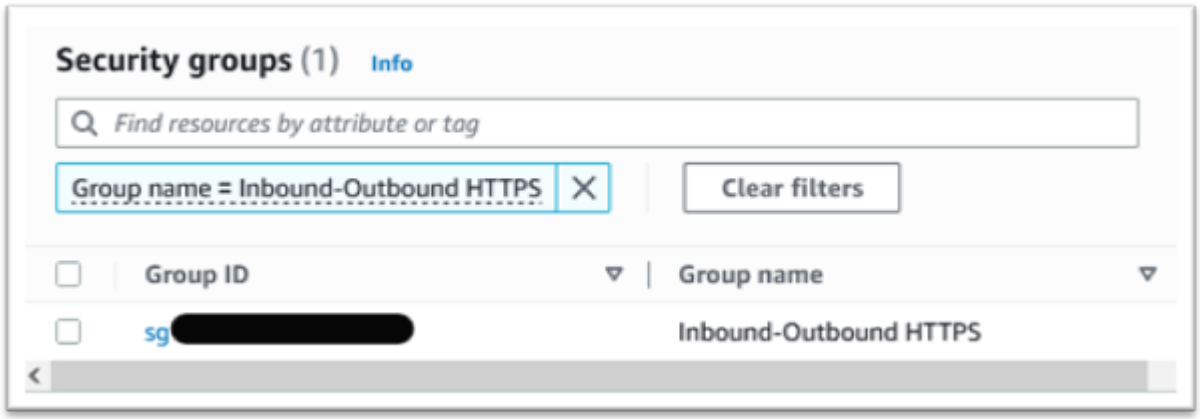

7. Wählen Sie für VPC die VPC für die Instance aus.




8. Wählen Sie die Availability Zone und die Subnetze für die VPC aus.




9. Wählen Sie die zuvor erstellte Sicherheitsgruppe aus.




10. Wählen Sie unter Richtlinie die Option Vollzugriff aus.




11. Klicken Sie auf Endpunkt erstellen.
12. Wiederholen Sie diesen Vorgang für die übrigen Schnittstellen.

# Richten Sie die Automatisierung für Rocket Enterprise Analyzer (ehemals Micro Focus) und Rocket Enterprise Developer Streaming-Sitzungen ein

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 zur Verfügung stellt, 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 zur Verfügung stellt, 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 in Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer) als Tabellen und Spalten anzeigen

Sie können mithilfe der Runtime von Rocket Software (ehemals Micro Focus) auf AWS Mainframe-Datensätze zugreifen, die im Rahmen von Mainframe Modernization bereitgestellt werden. Sie können die migrierten Datensätze als Tabellen und Spalten von einer Rocket Enterprise Developer-Instanz aus anzeigen. Wenn Sie Datensätze auf diese Weise betrachten, 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 Rocket Software-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: Datensätze von Rocket Software (ehemals Micro Focus) als Tabellen und Spalten anzeigen


## Voraussetzungen

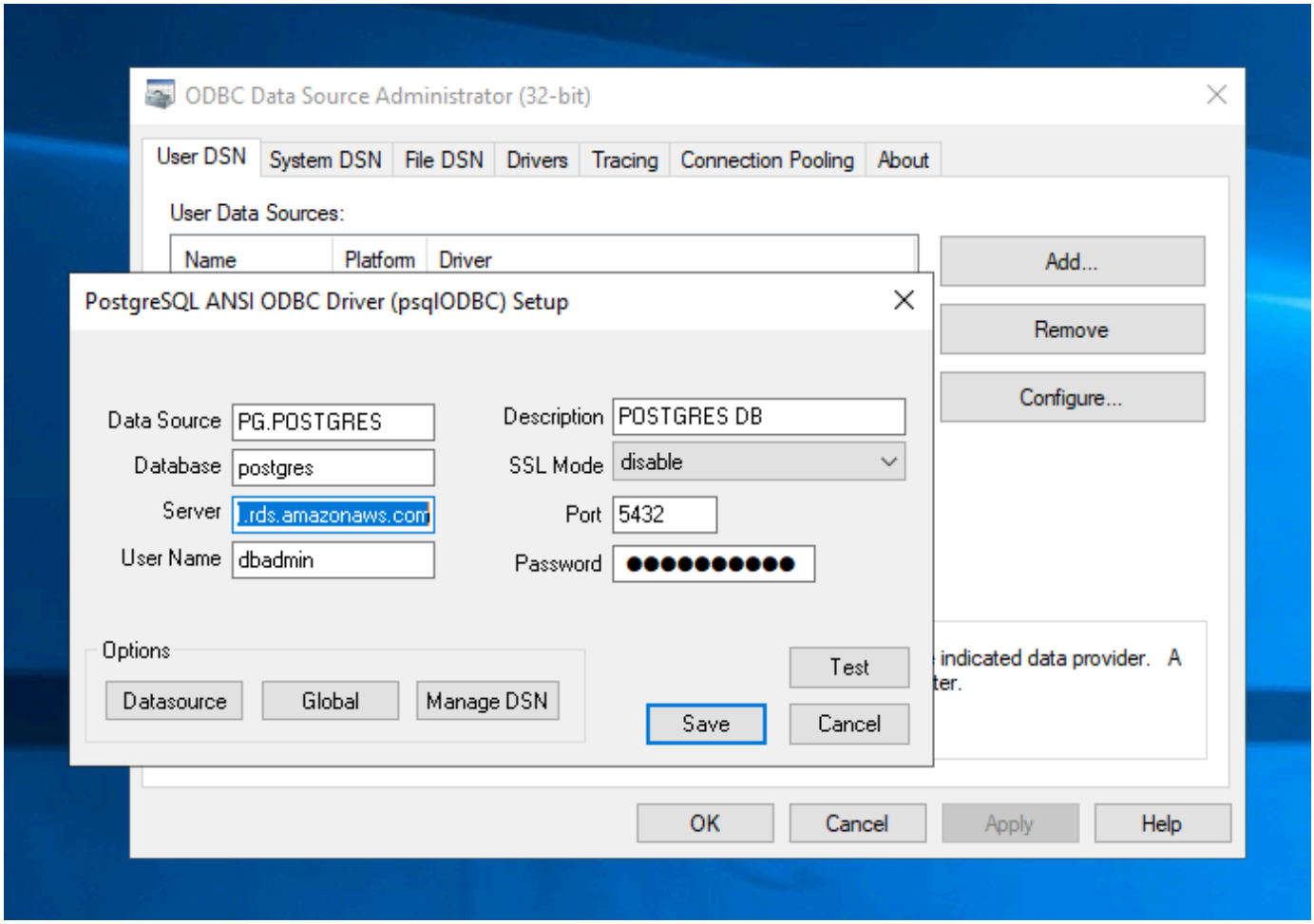
- Sie müssen über AppStream 2.0 Zugriff auf Rocket Enterprise Developer Desktop haben.
- Sie müssen eine Anwendung im Rahmen der AWS Mainframe-Modernisierung mithilfe der Runtime-Engine von Rocket Software bereitgestellt und ausgeführt haben.
- Sie speichern Ihre Anwendungsdaten in der Aurora PostgreSQL-kompatiblen Edition.

## Schritt 1: ODBC-Verbindung zum Rocket Software-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 Rocket 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.
4. 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
```



5. Wählen Sie Test, um sicherzustellen, dass die Verbindung funktioniert. Sie sollten die Meldung sehen, `Connection successful` ob der Test erfolgreich war.

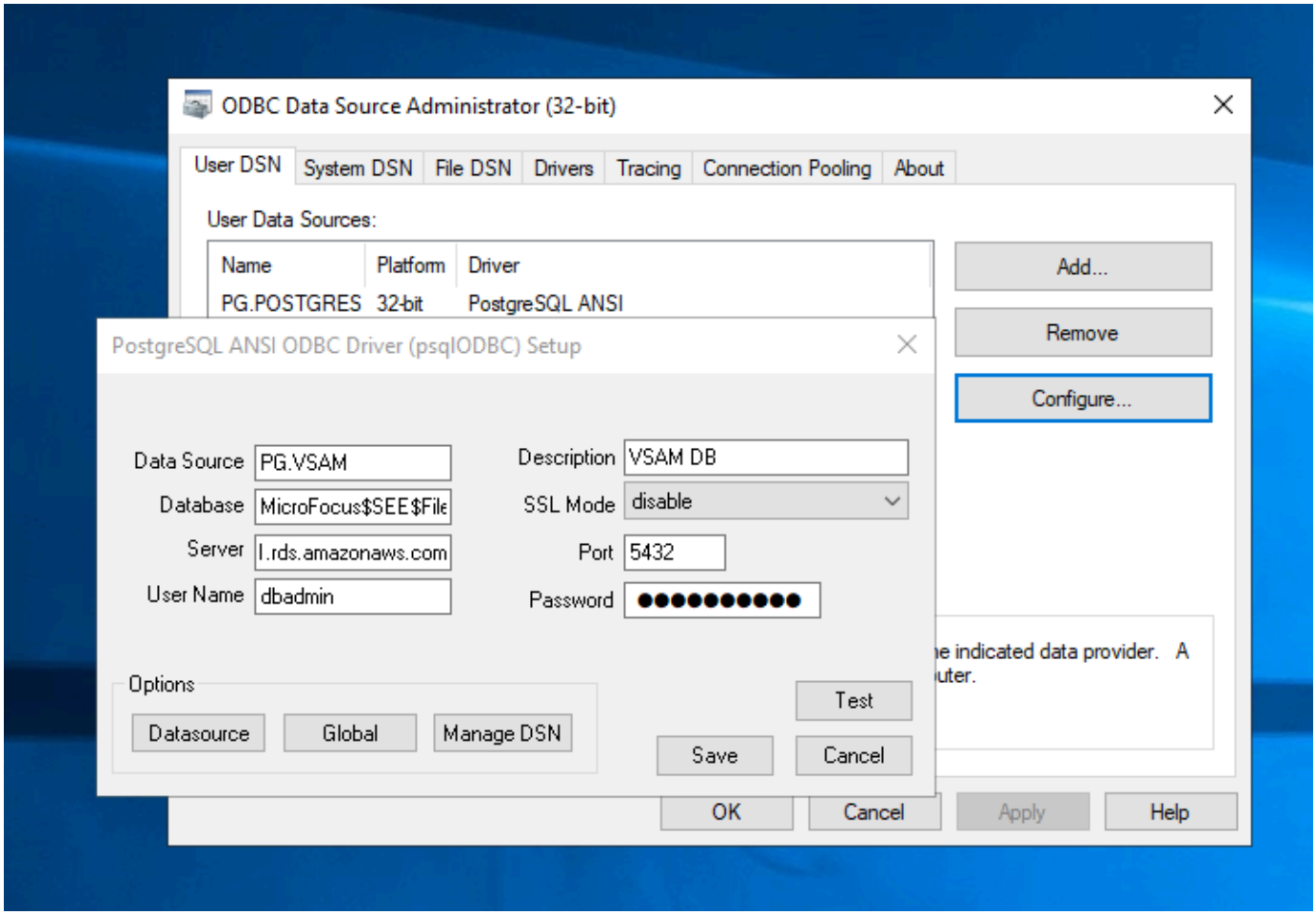
   Wenn der Test nicht erfolgreich ist, überprüfen Sie die folgenden Informationen.

   - Fehlerbehebung für Amazon RDS
   - Wie löse ich Probleme beim Herstellen einer Verbindung zu meiner Amazon RDS-DB-Instance?

6. Speichern Sie die Datenquelle.

7. 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.

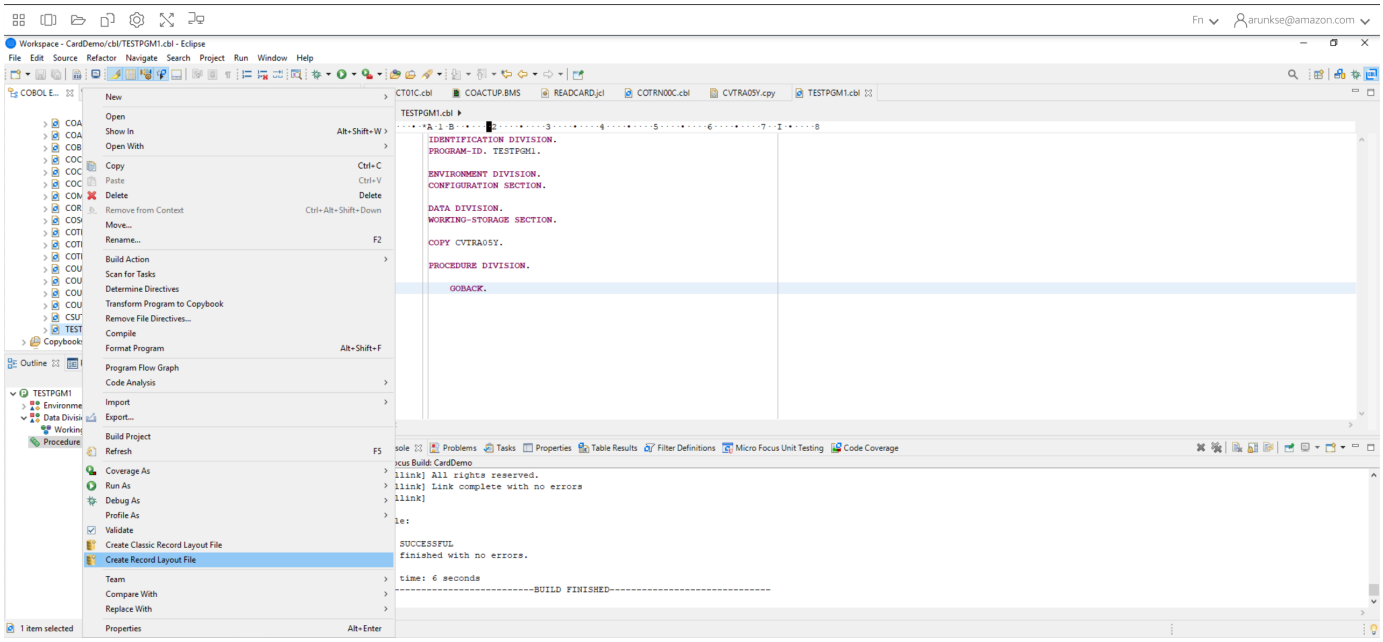
      DATA DIVISION.
      WORKING-STORAGE SECTION.

      COPY CVTRA05Y.

      PROCEDURE DIVISION.

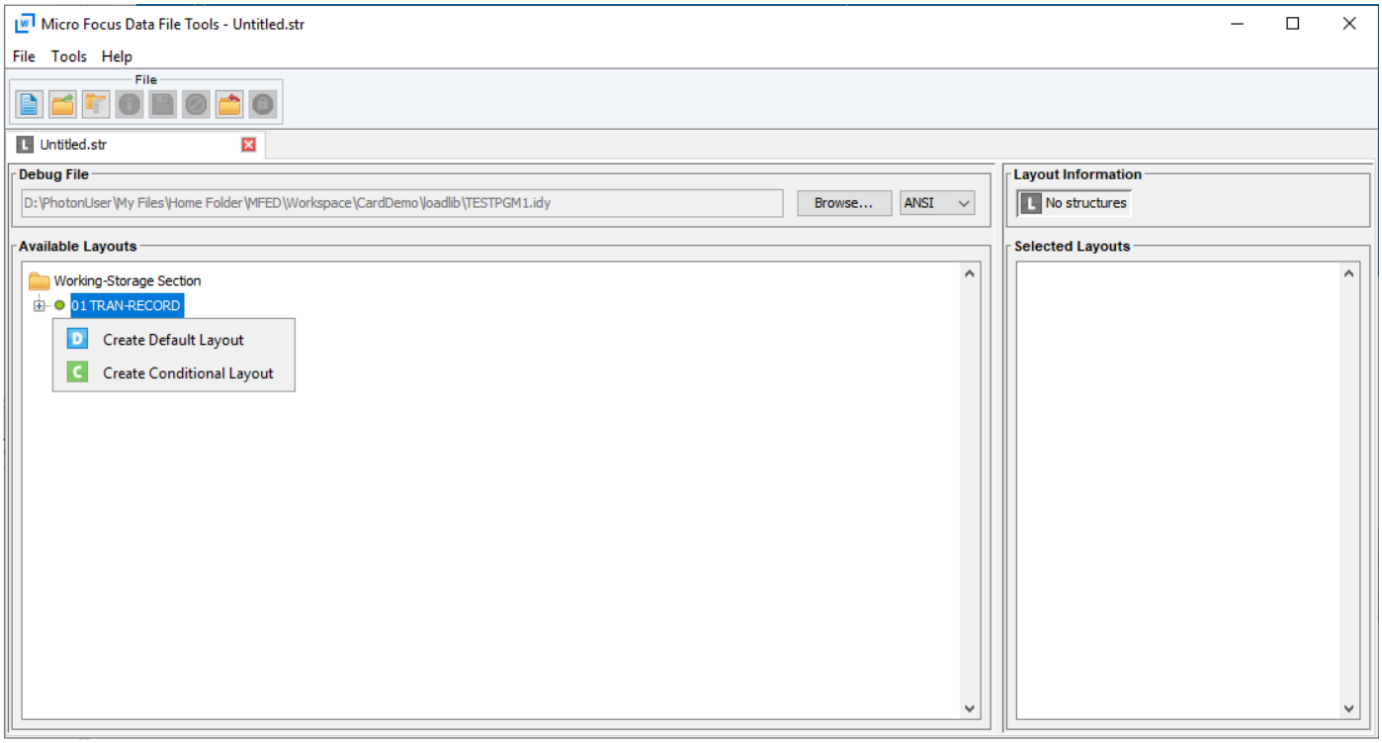
      GOBACK.
```

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.

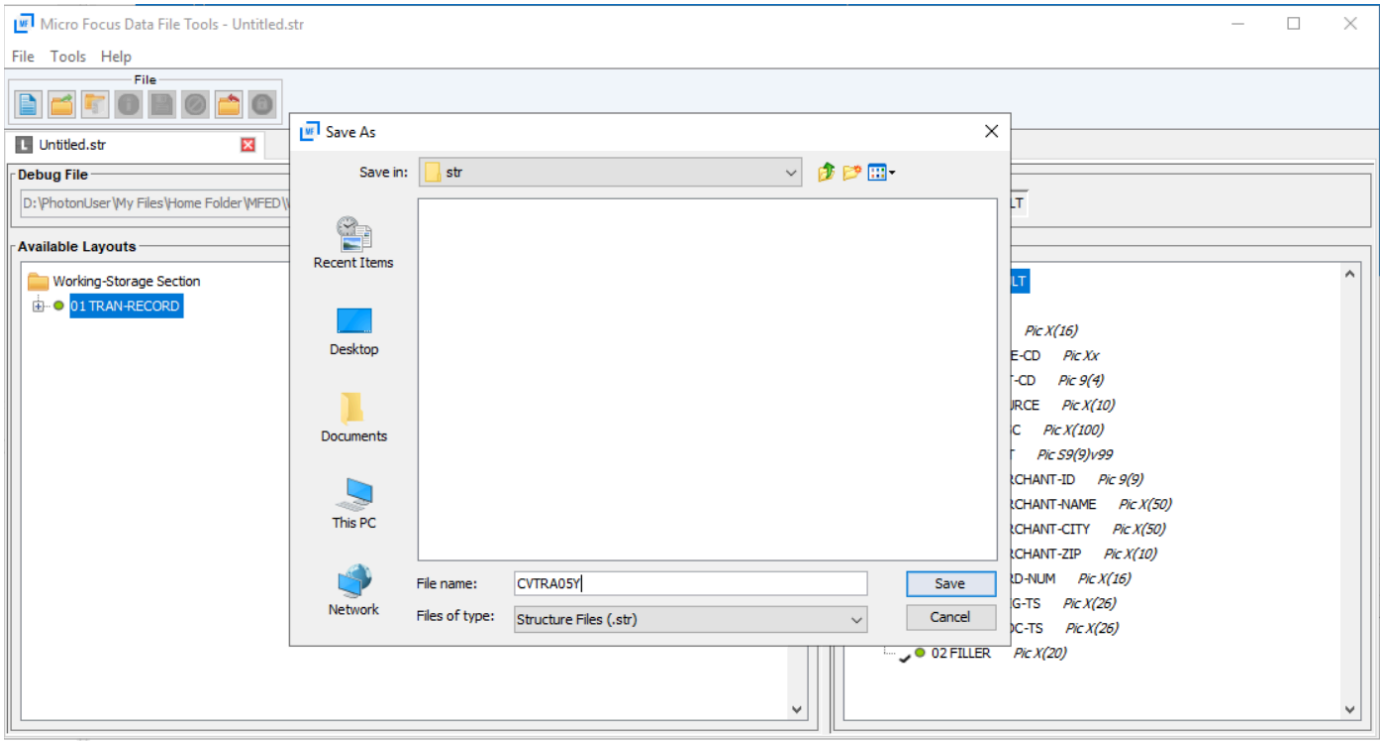



3. Klicken Sie mit der rechten Maustaste auf die Datensatzstruktur und wählen Sie je nach Layout Standardlayout erstellen (einzelne Struktur) oder Bedingtes Layout erstellen (mehrere Strukturen).

   Weitere Informationen finden Sie unter Strukturdateien und Layouts erstellen in der Micro Focus-Dokumentation.



4. Nachdem Sie das Layout erstellt haben, wählen Sie im Menü die Option 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.

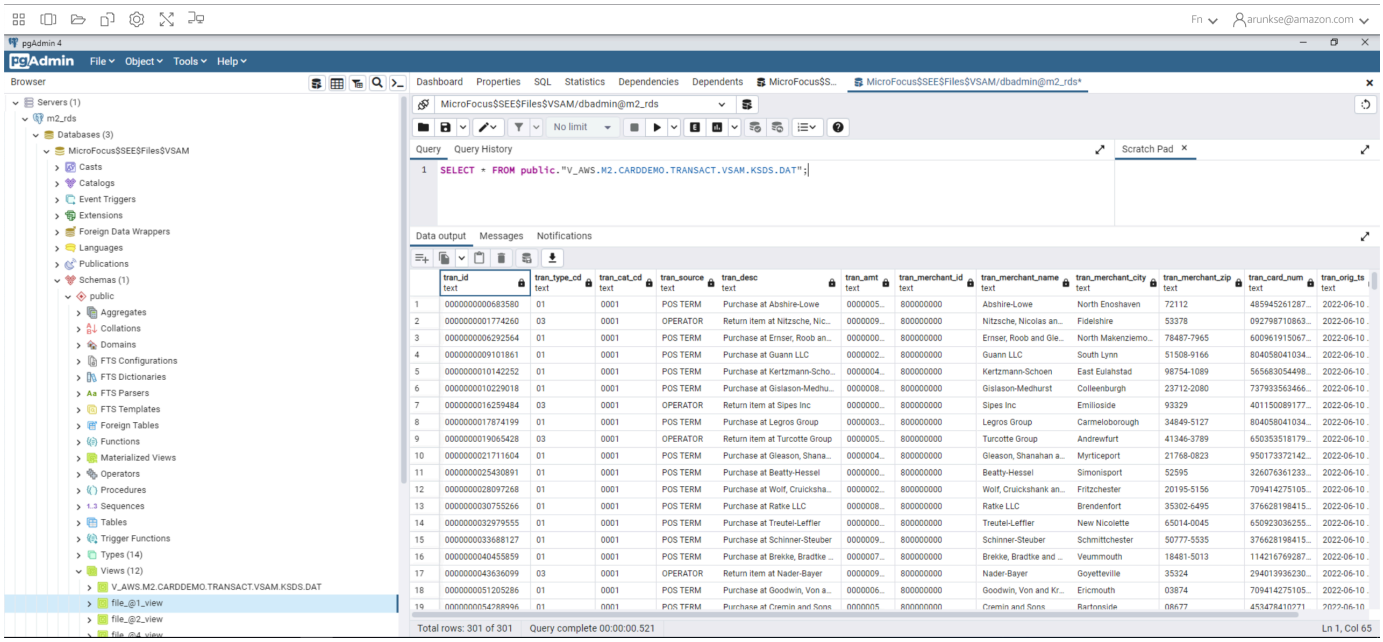
      VGN0017I Using structure definition 'TRAN-RECORD-DEFAULT'
      VGN0022I View 'V_AWS.M2.CARDDEMO.TRANSACT.VSAM.KSDS.DAT' installed in
 datastore 'sql://espacdatabase/VSAM'
      VGN0002I The operation completed successfully
```

## Schritt 5: Datensätze von Rocket Software (ehemals Micro Focus) 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.TRANSACT.VSAM.KSDS.DAT";
```




# Bearbeiten Sie Datensätze mit den Datendatei-Tools von Rocket Software (ehemals Micro Focus) in Enterprise Developer

Sie können Datensätze in AWS Mainframe Modernization mithilfe der Rocket Software-Runtime für alle migrierten Datensätze anzeigen und bearbeiten.
Die Schritte in diesem Dokument führen Sie durch den Prozess des Zugriffs auf Datensätze mithilfe von Data File Tools.
Auf diese Weise können Sie die migrierten Datensätze nach Bedarf anzeigen und bearbeiten.

Themen

- Voraussetzungen
- Starten Sie die Data File Tools von Rocket Software (ehemals Micro Focus)
- Bearbeiten Sie die in der MFDBFH-Datenbank gespeicherten VSAM-Datensätze
- Bearbeiten Sie Nicht-VSAM-Datensätze, die in der MFDBFH-Datenbank gespeichert sind
- Bearbeiten Sie im Dateisystem gespeicherte VSAM- und Nicht-VSAM-Datensätze (EFS/FSx)

## Voraussetzungen

Bevor Sie beginnen, müssen Sie eine Anwendung mit den Datensätzen bereitgestellt haben wurde im Rahmen des AWS Mainframe-Modernisierungsdienstes mithilfe der Rocket Software-Engine importiert.

Um mit der Bearbeitung der Datensätze fortzufahren, müssen Sie Schritt 1, Schritt 2 abschließen. und (optional) Schritt 3 von der the section called “Datensätze in Enterprise Developer als Tabellen anzeigen” Seite um die ODBC-Verbindung und den Micro Focus-Datenspeicher (das heißt,) zu konfigurieren. `MFDBFH`

> **Important**
>
> In diesem Handbuch wird davon ausgegangen, dass Sie Amazon Aurora Postgres als Micro Focus-Datenspeicher verwenden () `MFDBFH` um Ihre Anwendungsdaten zu speichern.

## Starten Sie die Data File Tools von Rocket Software (ehemals Micro Focus)

Nachdem Sie die Voraussetzungen erfüllt haben, starten Sie die Micro Focus Data File Tools, indem Sie die `MFDBFH_CONFIG` Umgebungsvariable für den Zugriff auf die in der Datenbank gespeicherten Datensätze einrichten (`MFDBFH`).

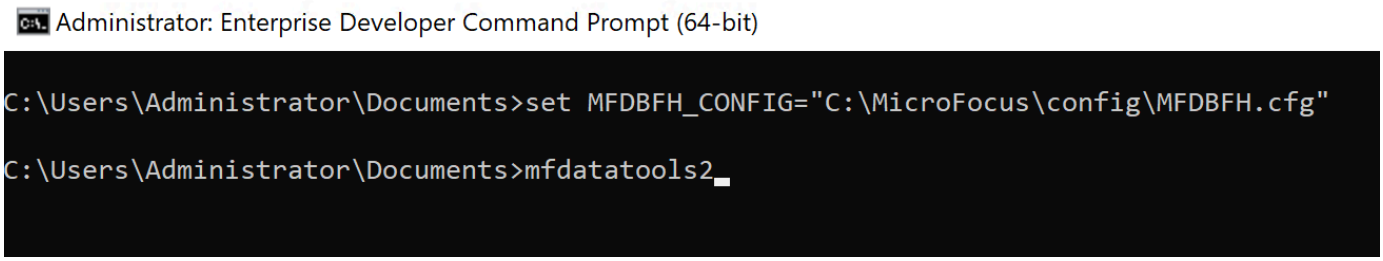
Gehen Sie dazu wie folgt vor:

1. Melden Sie sich beim Micro Focus Enterprise Developer-Desktop an und starten Sie die Enterprise Developer-Eingabeaufforderung (64-Bit) über das Startmenü.
2. Geben Sie der `MFDBFH_CONFIG` Umgebungsvariablen den vollständigen Pfad zu Ihrer `MFDBCH.cfg` Datei an.

   ```
   set MFDBFH_CONFIG="C:\MicroFocus\config\MFDBFH.cfg"
   ```

3. Starten Sie Micro Focus Data File Tools über die Enterprise Developer-Befehlszeile mit dem folgenden Befehl.

   ```
   mfdatatools2
   ```



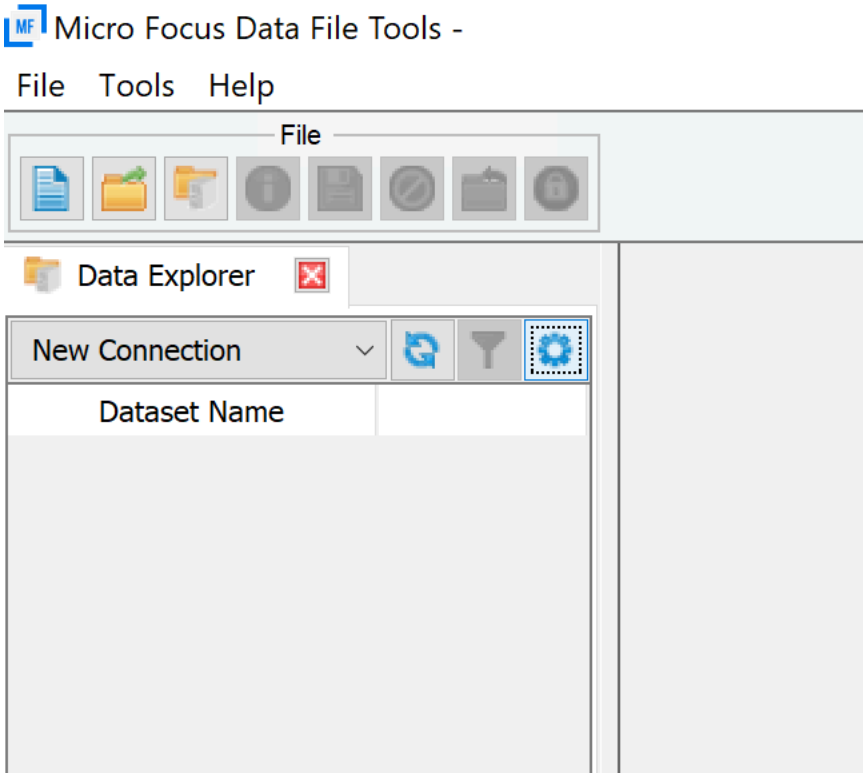

Dadurch werden die Micro Focus Data File Tools in einem separaten Fenster geöffnet.

## Bearbeiten Sie die in der MFDBFH-Datenbank gespeicherten VSAM-Datensätze

Sobald Sie die Micro Focus Data File Tools gestartet haben, öffnen Sie einen VSAM-Datensatz, der im Micro Focus-Datenspeicher gespeichert ist.

Gehen Sie dazu wie folgt vor:

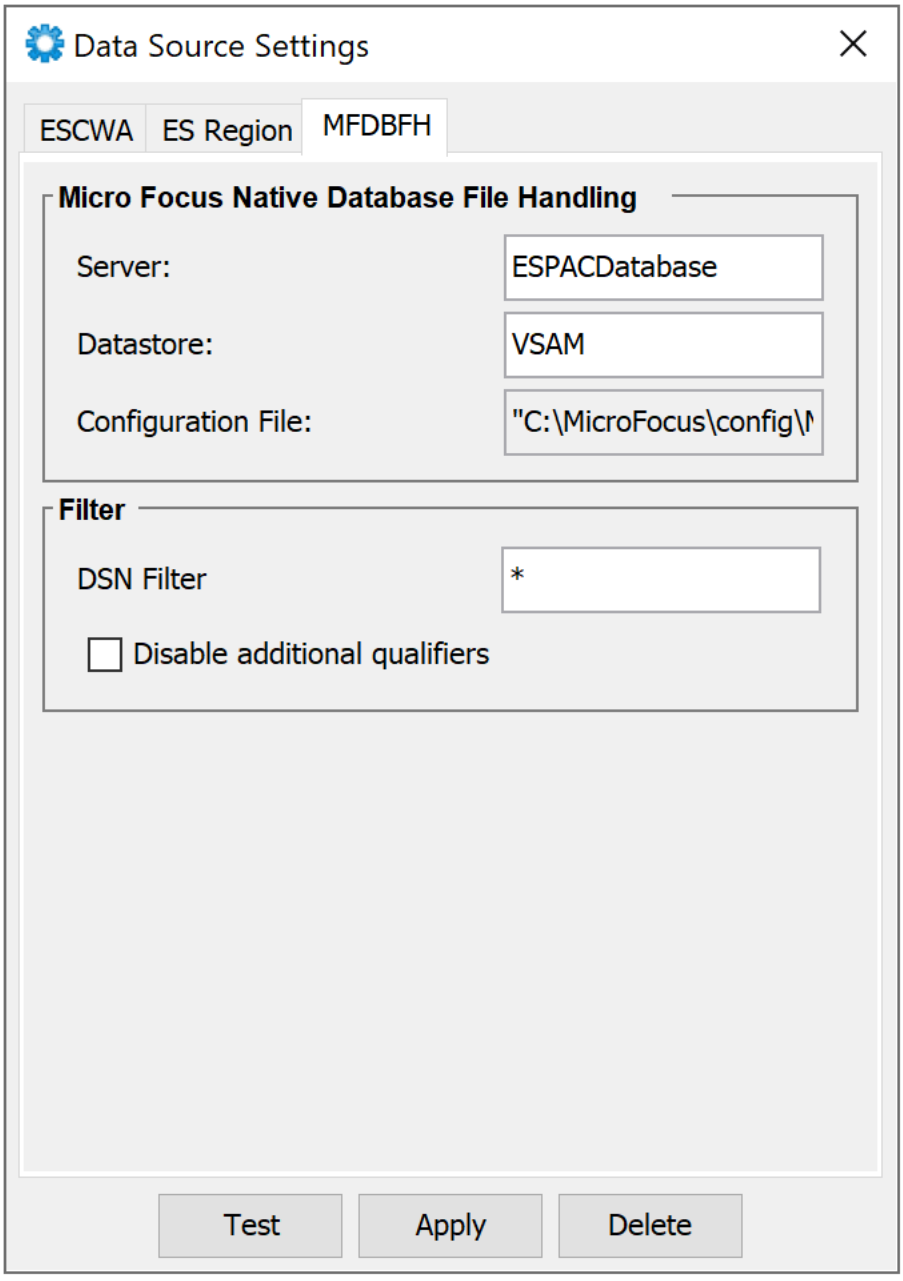
1. Wählen Sie im Fenster Micro Focus Data File Tools im Menü Datei die Option Data Explorer.
2. Wählen Sie im Bereich Data Explorer die Option Einstellungen (Zahnradsymbol), um eine neue Verbindung zu konfigurieren. Dadurch wird ein Fenster mit Datenquelleneinstellungen geöffnet.



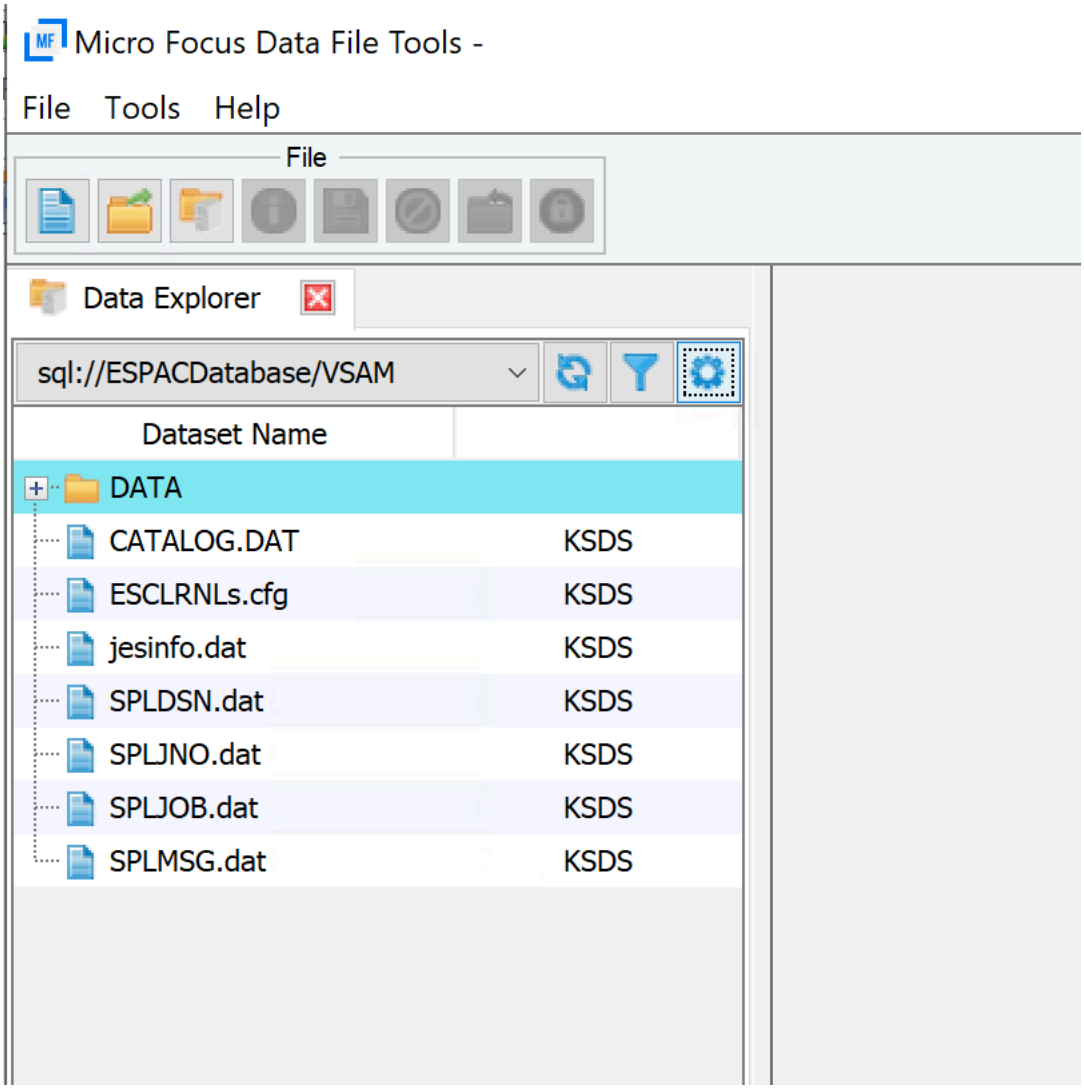
3. Wählen Sie im Fenster Datenquelleneinstellungen die Registerkarte MFDBFH und geben Sie die folgenden Werte ein:

   - Server: `ESPACDatabase`
   - Datenspeicher: `VSAM`

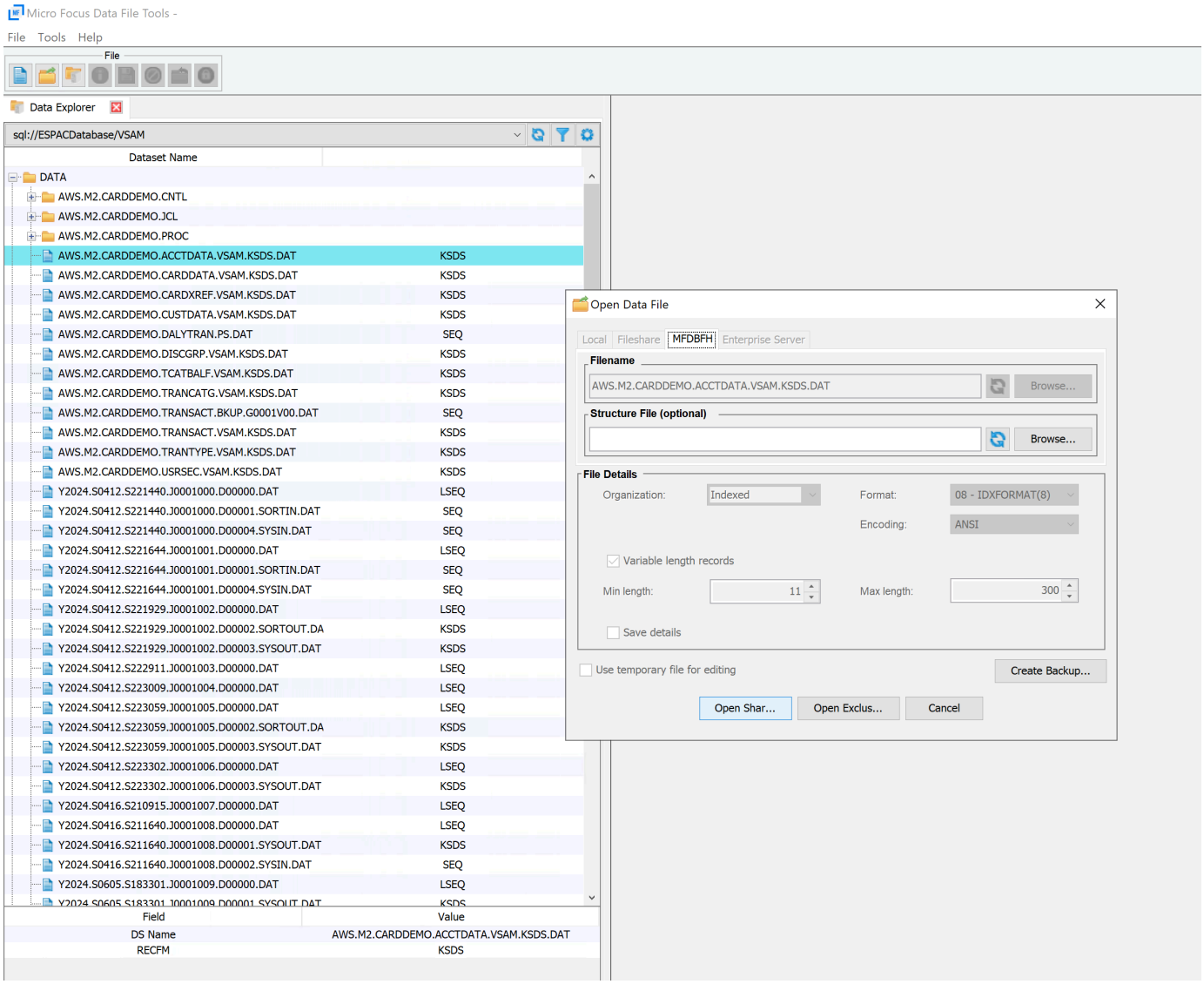
   Wählen Sie Anwenden, um die Konfiguration zu speichern.




   Der Datenexplorer zeigt jetzt alle Datensätze an, die in gespeichert sind. `MFDBFH`




4. Erweitern Sie den relativen Pfad `DATA` und doppelklicken Sie auf den VSAM-Datensatz, den Sie öffnen möchten.
5. Wählen Sie im Fenster „Datendatei öffnen“ die Option „Gemeinsam öffnen“ oder „Exklusiv öffnen“, um den Datensatz zu öffnen.




Sie können den geöffneten Datensatz jetzt anzeigen oder bearbeiten.

## Bearbeiten Sie Nicht-VSAM-Datensätze, die in der MFDBFH-Datenbank gespeichert sind

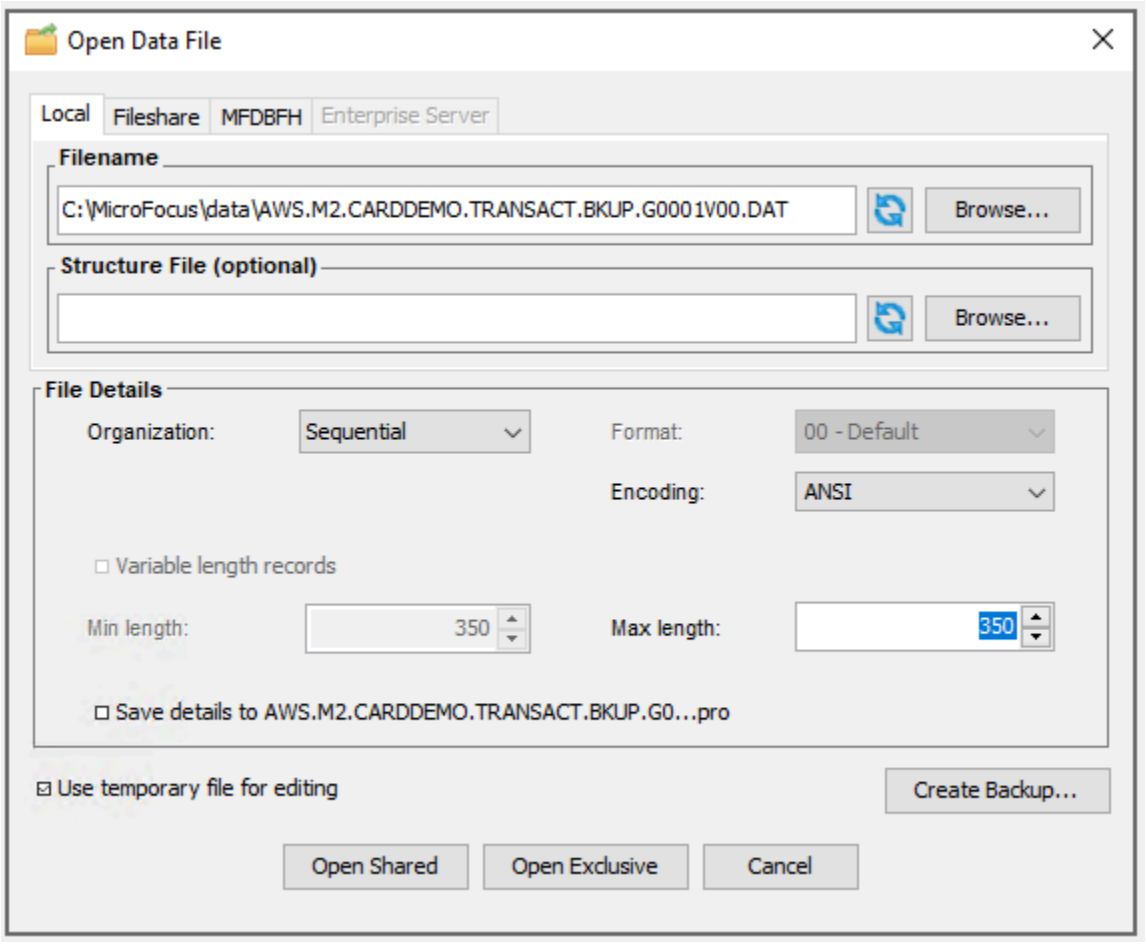
Wenn Sie Nicht-VSAM-Datensätze bearbeiten möchten, öffnen Sie einen Nicht-VSAM-Datensatz, der im Micro Focus-Datenspeicher gespeichert ist.

Gehen Sie dazu wie folgt vor:

1. Führen Sie an der Enterprise Developer-Befehlszeile (64-Bit) den `dbfhdeploy data extract` Befehl aus, um den Nicht-VSAM-Datensatz in Ihr lokales Dateisystem herunterzuladen.

> **Note**
>
> Bevor Sie diesen Befehl ausführen, stellen Sie sicher, dass Sie der `MFDBFH_CONFIG` Umgebungsvariablen den vollständigen Pfad zu Ihrer `MFDBFH.cfg` Datei zugewiesen haben.
>
> ```
> dbfhdeploy data extract sql://ESPACDatabase/VSAM/
> AWS.M2.CARDDEMO.TRANSACT.BKUP.G0001V00.DAT?folder=/DATA C:\MicroFocus\data
> \AWS.M2.CARDDEMO.TRANSACT.BKUP.G0001V00.DAT
> ```

2. Starten Sie die Micro Focus Data File Tools über das Startmenü.
3. Wählen Sie im Menü Datei der Micro Focus Data File Tools die Option Öffnen und dann Datendatei.
4. Durchsuchen Sie im Fenster „Datendatei öffnen“ den heruntergeladenen Datensatz in Ihrem lokalen Dateisystem. Bearbeiten Sie die Dateidetails nach Bedarf. Wählen Sie dann „Gemeinsam öffnen“ oder „Exklusiv öffnen“, um den Datensatz zu öffnen.




Sie können den geöffneten Datensatz jetzt anzeigen oder bearbeiten.

Die bearbeiteten oder aktualisierten Datensätze können mithilfe der Schritte unter the section called “Datensätze für Anwendungen importieren” oder mithilfe des Befehlszeilendienstprogramms dbfhdeploy wieder in den Micro Focus-Datenspeicher importiert werden.

## Bearbeiten Sie im Dateisystem gespeicherte VSAM- und Nicht-VSAM-Datensätze (EFS/FSx)

Sie können auch einen Datensatz öffnen, der in einem Dateisystem gespeichert ist.

Gehen Sie dazu wie folgt vor:

1. Montieren Sie den EFS/FSx Dateisystem in der Enterprise EC2 Developer-Instanz.
2. Verwenden Sie die Micro Focus Data File Tools, um die Datensätze im Dateisystem zu durchsuchen und zu öffnen.

# Tutorials für Rocket Software (ehemals Micro Focus)

Die Tutorials in diesem Abschnitt helfen Ihnen beim Einstieg in die Einrichtung verschiedener Aufgaben in der Runtime-Engine von Rocket Software für den AWS Mainframe Modernization Service. Diese Tutorials dienen zum Einrichten einer Beispielanwendung, CI/CD Pipelines, die Verwendung von Vorlagen mit Rocket Enterprise Developer und die Einrichtung von Enterprise Analyzer.

Themen


- Tutorial: Den Build von Rocket Software (ehemals Micro Focus) für die BankDemo Beispielanwendung einrichten
- Tutorial: Einrichtung eines CI/CD Pipeline zur Verwendung mit Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer)
- Tutorial: AppStream 2.0 für die Verwendung mit Rocket Enterprise Analyzer und Rocket Enterprise Developer einrichten
- Tutorial: Vorlagen mit Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer) verwenden
- Tutorial: Enterprise Analyzer auf AppStream 2.0 einrichten
- Tutorial: Rocket Enterprise Developer auf AppStream 2.0 einrichten


## Tutorial: Den Build von Rocket Software (ehemals Micro Focus) für die BankDemo Beispielanwendung einrichten

AWS Die Mainframe-Modernisierung bietet Ihnen die Möglichkeit, Builds (und kontinuierlicheintegration/continuous delivery (CI/CD) Pipelines für Ihre migrierten Anwendungen einzurichten. Diese Builds und Pipelines verwenden und AWS CodeBuild AWS CodeCommit, AWS CodePipeline um diese Funktionen bereitzustellen. 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.

In diesem Tutorial wird gezeigt, 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: Teilen Sie die Build-Assets mit AWS Ihrem Konto
- Schritt 2: Amazon S3 S3-Buckets erstellen
- Schritt 3: Erstellen Sie die Build-Spezifikationsdatei
- Schritt 4: Laden Sie die Quelldateien hoch
- Schritt 5: IAM-Richtlinien erstellen
- Schritt 6: Erstellen Sie eine IAM-Rolle
- Schritt 7: Hängen Sie die IAM-Richtlinien an die IAM-Rolle an
- Schritt 8: Erstellen Sie das CodeBuild Projekt
- Schritt 9: Starten Sie den Build
- Schritt 10: 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 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: Teilen Sie die Build-Assets mit AWS Ihrem Konto

In diesem Schritt stellen Sie sicher, dass Sie die Build-Assets mit Ihrem AWS Konto teilen, insbesondere in der Region, in der die Assets verwendet werden.

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.

> **Important**
>
> Sie müssen diesen Schritt in jeder AWS Region, in der Sie Builds erstellen möchten, einmal ausführen.

## Schritt 2: 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 3: 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 zum Ausführen des Builds. 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 die Datei`BankDemo.zip`.

## Schritt 4: 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 Upload.

## Schritt 5: 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 Build-Tools von Rocket Software 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-*-<region>-prod"
        }
    ]
}
```

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 Richtlinien-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 6: 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 Weiter aus.
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 7: 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 8: 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, `codebuild-regionId-accountId-input-bucket` z. B.
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 Rocket Software 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 Rocket Software 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 Rocket Software v8: Geben Sie ein `673918848628.dkr.ecr.us-west-2.amazonaws.com/m2-enterprise-build-tools:8.0.9.R1`
   - Für Rocket Software 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 Existierende Servicerolle aus und wählen Sie im Feld Rollen-ARN die Servicerolle aus, die Sie zuvor erstellt haben, `BankdemoCodeBuildServiceRole` z. B.
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 9: 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 10: Laden Sie die Ausgabeartefakte herunter

In diesem Schritt laden Sie die Ausgabeartefakte von Amazon S3 herunter. Das Build-Tool von Rocket Software 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 CodeBuildim AWS CodeBuild Benutzerhandbuch unter Löschen eines Build-Projekts.

## Tutorial: Einrichtung eines CI/CD Pipeline zur Verwendung mit Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer)

Dieses Tutorial zeigt Ihnen, wie Sie die BankDemo Beispielanwendung in Rocket Enterprise Developer importieren, bearbeiten, kompilieren und ausführen und anschließend Ihre Änderungen übernehmen, um eine CI/CD Pipeline.

Inhalt


- Voraussetzungen
- Erstellen CI/CD grundlegende Infrastruktur der Pipeline
- AWS CodeCommit Repository erstellen und 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
  - Starten Sie das Rumba 3270-Terminal
  - Führen Sie BankDemo eine Transaktion aus
  - Stoppen Sie den BANKDEMO-Server von Enterprise Developer
- 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: Übernehmen Sie die Änderungen und führen Sie den CI/CD Pipeline
- 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.yml`
  - 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.

Jede Datei hat folgenden Zweck:

`basic-infra.yaml`

Diese AWS CloudFormation Vorlage erstellt die grundlegende Infrastruktur, die benötigt wird für CI/CD Pipeline: 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 Verzeichnisstruktur`config_git.yaml`, und startet den Pipeline-Stack mithilfe von`pipeline.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 CI/CD grundlegende Infrastruktur der Pipeline

Verwenden Sie die AWS CloudFormation Vorlage`basic-infra.yaml`, um den grundlegenden Infrastrukturstapel der CI/CD-Pipeline über die AWS CloudFormation Konsole zu erstellen. 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 /cloudformation an AWS Management Console und öffnen Sie die AWS CloudFormation Konsole. https://console.aws.amazon.com
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 URLs je nach Ihrer Vorlage eine der folgenden Optionen 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

m2-cicd-pipeline-foundation

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.

No (Create one)

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.

11.11.0.0/16

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.

5

### 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.

m2

Name of the source package file for the infrastructure Lambda function

Don't change unless you know what you are doing.

cicd/mf/m2-code-sync-function.zip

Full URL of the pipeline CloudFormation template file

Don't change unless you know what you are doing.

https://m2-us-east-1.s3.us-east-1.amazonaws.com/cicd/mf/pipeline.yaml

What name prefix to use for the new S3 buckets?

A name prefix for the S3 buckets that will be created by this stack.

mf

Cancel Previous Next

Nehmen Sie die folgenden Änderungen vor:

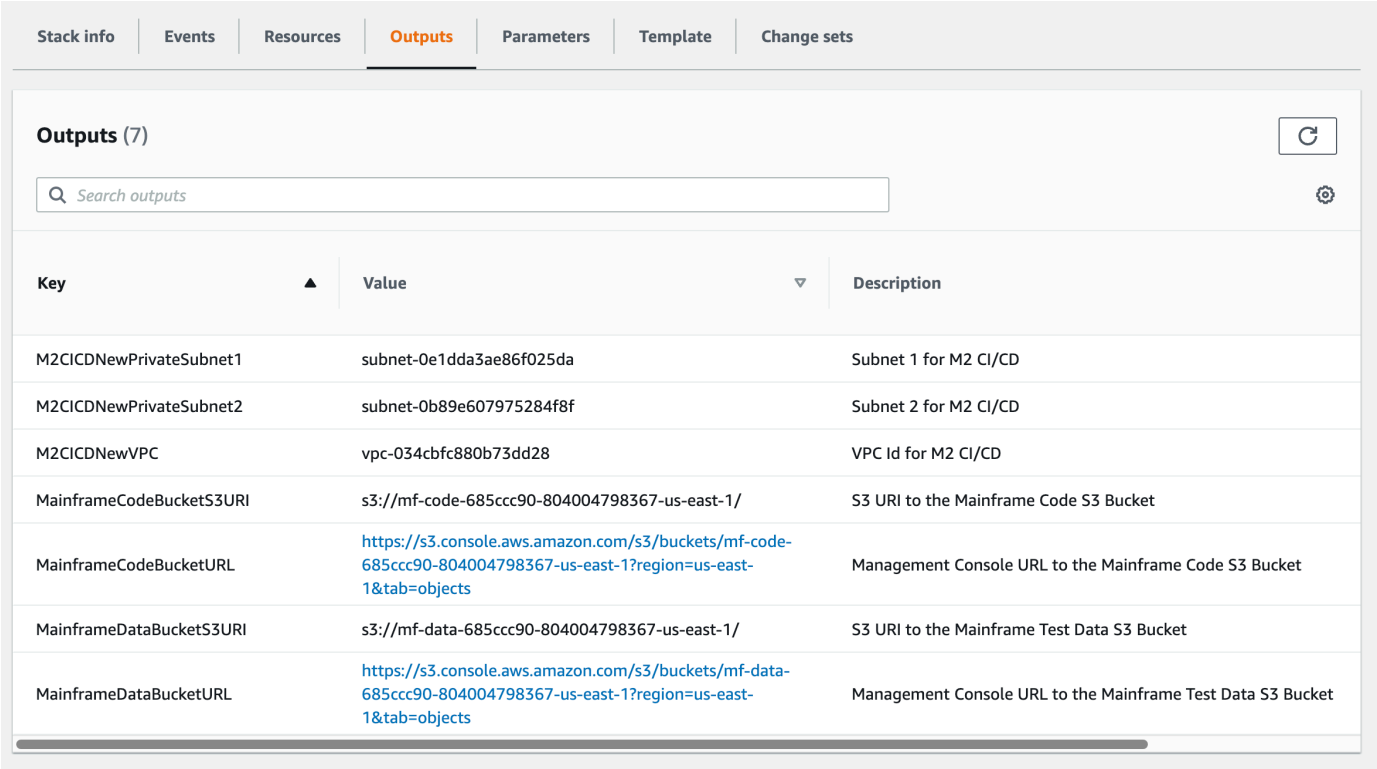
- Geben Sie die entsprechenden Werte für den Stack-Namen und die Parameter für die Netzwerkkonfiguration an.
- Die meisten Parameter in Bereitstellungskonfigurationen sind entsprechend vorab ausgefüllt, sodass Sie sie nicht ändern müssen. Ändern Sie je nach Ihrem AWS-Region die AWS CloudFormation Pipeline-Vorlage in eine der folgenden Amazon S3 S3-Vorlagen 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.




## AWS CodeCommit Repository erstellen und CI/CD Pipeline

In diesem Schritt erstellen Sie ein CodeCommit Repository und stellen ein bereit CI/CD Pipeline-Stack durch Aufrufen einer Lambda-Funktion, 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 wurdeErstellen CI/CD grundlegende Infrastruktur der 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
   ```

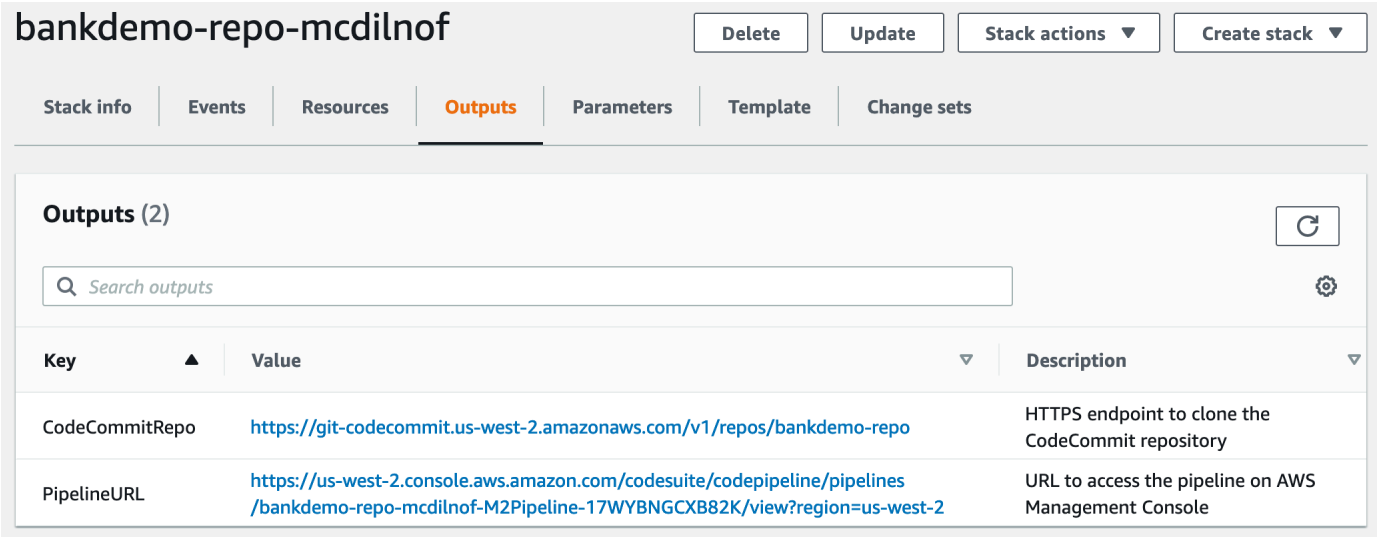
   - Fügen Sie die E-Mail-Adresse hinzu, an die Sie Benachrichtigungen erhalten möchten.

```
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
```

5. Laden Sie die `config_git.yml` Datei mit der Definition der CodeCommit Repository-Ordnerstruktur in den Amazon S3 S3-Bucket hoch, der in erstellt wurdeErstellen CI/CD grundlegende Infrastruktur der 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 die 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. `bankdemo-repo-`*`01234567`* B. Sie finden die CodeCommit Repository-URL und die URL für den Zugriff auf die erstellte Pipeline in der AWS Management Console.




6. Wenn die Erstellung des CodeCommit Repositorys abgeschlossen ist, CI/CD Die Pipeline wird sofort ausgelöst, um eine vollständige CI/CD.

7. Sobald die Datei 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 Rocket Enterprise Developer auf AppStream 2.0 finden Sie unterTutorial: Rocket Enterprise Developer auf AppStream 2.0 einrichten.

Um das CodeCommit Repository mit Enterprise Developer zu verbinden, verwenden Sie den `target-repository` in angegebenen NamenBeispiel 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
- Starten Sie das Rumba 3270-Terminal
- Führen Sie BankDemo eine Transaktion aus
- Stoppen Sie den BANKDEMO-Server von Enterprise Developer

Connect zu der Enterprise Developer AppStream 2.0-Instanz her, in der Sie sie erstellt habenErstellung 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 Folgendes ein `C:\Users\<username>\workspace` 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 Arbeitsbereich 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\<username>\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 anschließend 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:
     - Dateiname:`D:\<username>\workspace\BANKDEMO\config\BANKDEMO_ED.json`.
     - Klicken Sie auf Open.
   - Wählen Sie Weiter aus.
   - 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

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 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 ein`BANK`.
2. Auf Bildschirm BANK10, im Eingabefeld für Benutzer-ID... :, geben Sie ein `guest` und drücken Sie die Eingabetaste.
3. Geben Sie auf Bildschirm BANK20 in das Eingabefeld vor Kreditkosten berechnen den Text `/` (Schrägstrich) ein und drücken Sie die Eingabetaste.
4. Auf Bildschirm BANK70:

   - In dem Betrag, den Sie leihen möchten... :, eingeben`10000`.
   - In Bei einem Zinssatz von... :, geben Sie ein`5.0`.
   - In Für wie viele Monate... :, eingeben`10`.
   - 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

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 „Kreditberechnung“ ein neues Feld BANK7 0 einrichten, in dem der gesamte Kreditbetrag angezeigt wird. Dies erfordert eine Änderung des BMS-Bildschirms MBANK7 0.CBL, das Hinzufügen eines neuen Felds und des entsprechenden Bildschirmverwaltungsprogramms SBANK7 0P.CBL mit zugehörigen Copybooks. Außerdem muss die Kreditberechnungsroutine in BBANK7 0P.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 Rocket 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 Analyseregel 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 SBANK70P.CBL und wählen Sie mit einem Doppelklick beliebige 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 MBANK7 0.BMS und SBANK7 0P.CBL (optional) *.

Kopieren Sie für schnelle Änderungen die folgenden Dateien:

1. ..\BANKDEMO-exercise\Exercis01\screens\MBANK70.BMS auf D:\PhotonUser\workspace\bankdemo\source\screens.
2. .\BANKDEMO-exercise\Exercis01\cobol\SBANK70P.CBL auf D:\PhotonUser\workspace\bankdemo\source\cobol.
3. ..\BANKDEMO-exercise\Exercis01\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.../CleanAlle Projekte aus.

Gehen Sie wie folgt vorSBANK70P.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
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

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 BBANK7 0P.CBL (optional) *.

Kopieren Sie zur schnellen Änderung die folgende Datei:

- `..\BANKDEMO-exercise\Exercis01\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
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

Schritt 4: Bestätigen Sie die Änderungen und führen Sie die CI/CD-Pipeline aus

Übernehmen Sie die Änderungen in das zentrale CodeCommit Repository und lösen Sie das aus CI/CD Pipeline zum Erstellen, Testen und Bereitstellen der Änderungen.

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: Übernehmen Sie die Änderungen und führen Sie den CI/CD Pipeline

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

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

> **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\cobol\BBANK70P.CBL` bis kopieren`D:\PhotonUser\workspace\bankdemo\source\cobol`.

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
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

> **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\cobol\BBANK70P.CBL` und `LOANCALC.CBL` nach kopieren`D:\PhotonUser\workspace\bankdemo\source\cobol`.

Schritt 3: Übernehmen Sie die Änderungen und führen Sie den CI/CD Pipeline

Übernehmen Sie die Änderungen in das zentrale CodeCommit Repository und lösen Sie das aus CI/CD Pipeline zum Erstellen, Testen und Bereitstellen der Änderungen.

1. Wählen Sie im BANKDEMO-Projekt im Kontextmenü die Option Team/Commit aus.
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.

## Tutorial: AppStream 2.0 für die Verwendung mit Rocket Enterprise Analyzer und Rocket 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 Repositorys 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 beschreibt, wie Amazon AppStream 2.0-Flotten und -Stacks eingerichtet werden, um die im Rahmen der AWS Mainframe-Modernisierung verwendeten Tools Rocket Enterprise Analyzer und Rocket Enterprise Developer zu hosten. Rocket Enterprise Analyzer wird normalerweise während der Assess-Phase und Rocket Enterprise Developer normalerweise während der Migrate- und Modernisierungsphase des Mainframe-Modernisierungsansatzes verwendet. AWS 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 Stapel mithilfe der AWS CloudFormation Vorlage
- 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- .yml. appstream-fleet-ea-ed
- 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  EC2 Amazon-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.

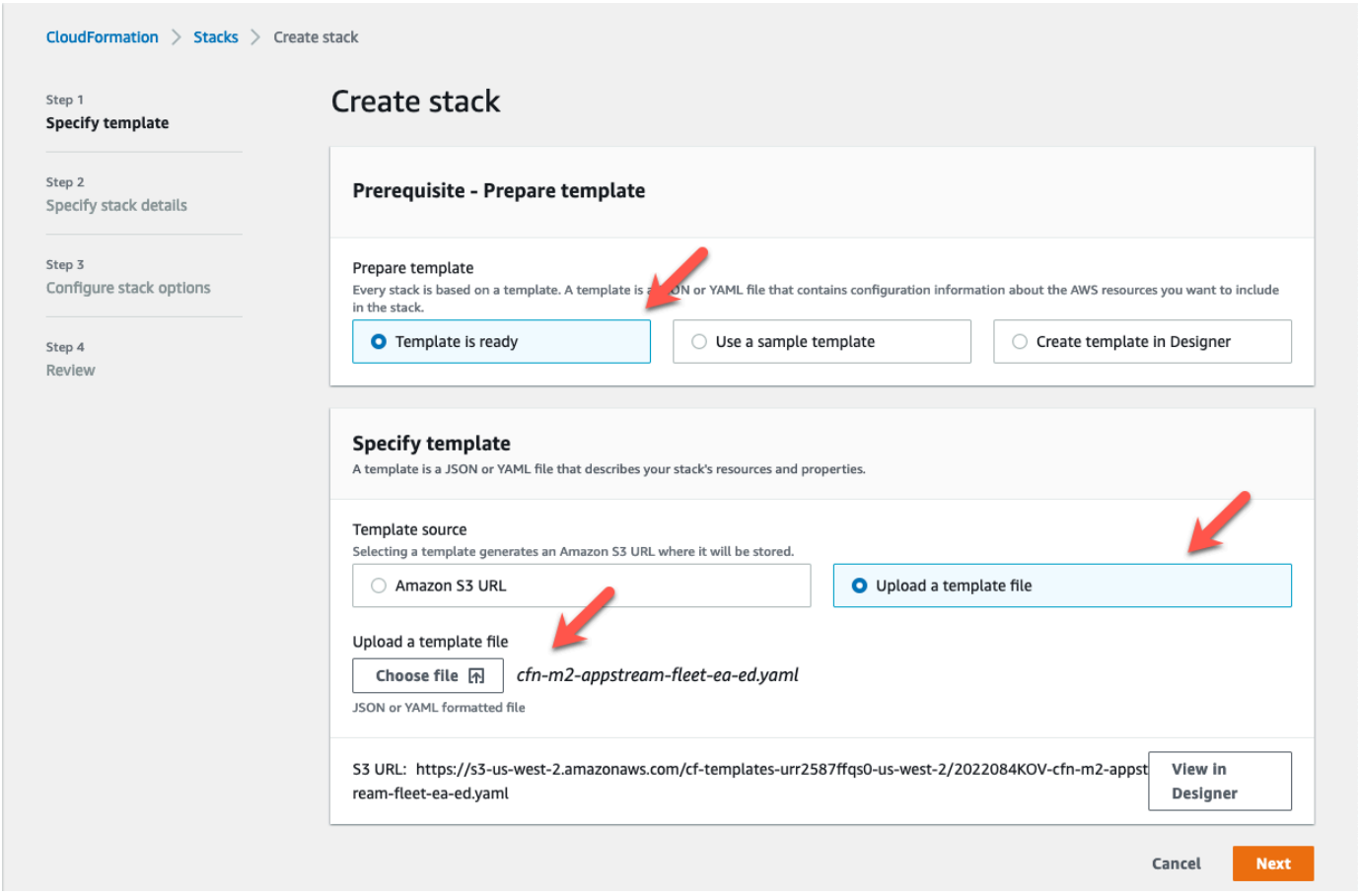
1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole unter https://console.aws.amazon.com/m2/.
2. Wählen Sie in der linken Navigationsleiste Tools aus.
3. Wählen Sie unter Assets analysieren, entwickeln und erstellen die Option Assets mit meinem AWS Konto teilen aus.

## Schritt 2: Erstellen Sie den Stapel mithilfe der AWS CloudFormation Vorlage

In diesem Schritt verwenden Sie die heruntergeladene AWS CloudFormation Vorlage, um einen AppStream 2.0-Stack und eine Flotte für den Betrieb von Rocket Enterprise Analyzer zu erstellen. Sie können diesen Schritt später wiederholen, um einen weiteren AppStream 2.0-Stack und eine Flotte für die Ausführung von Rocket 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.

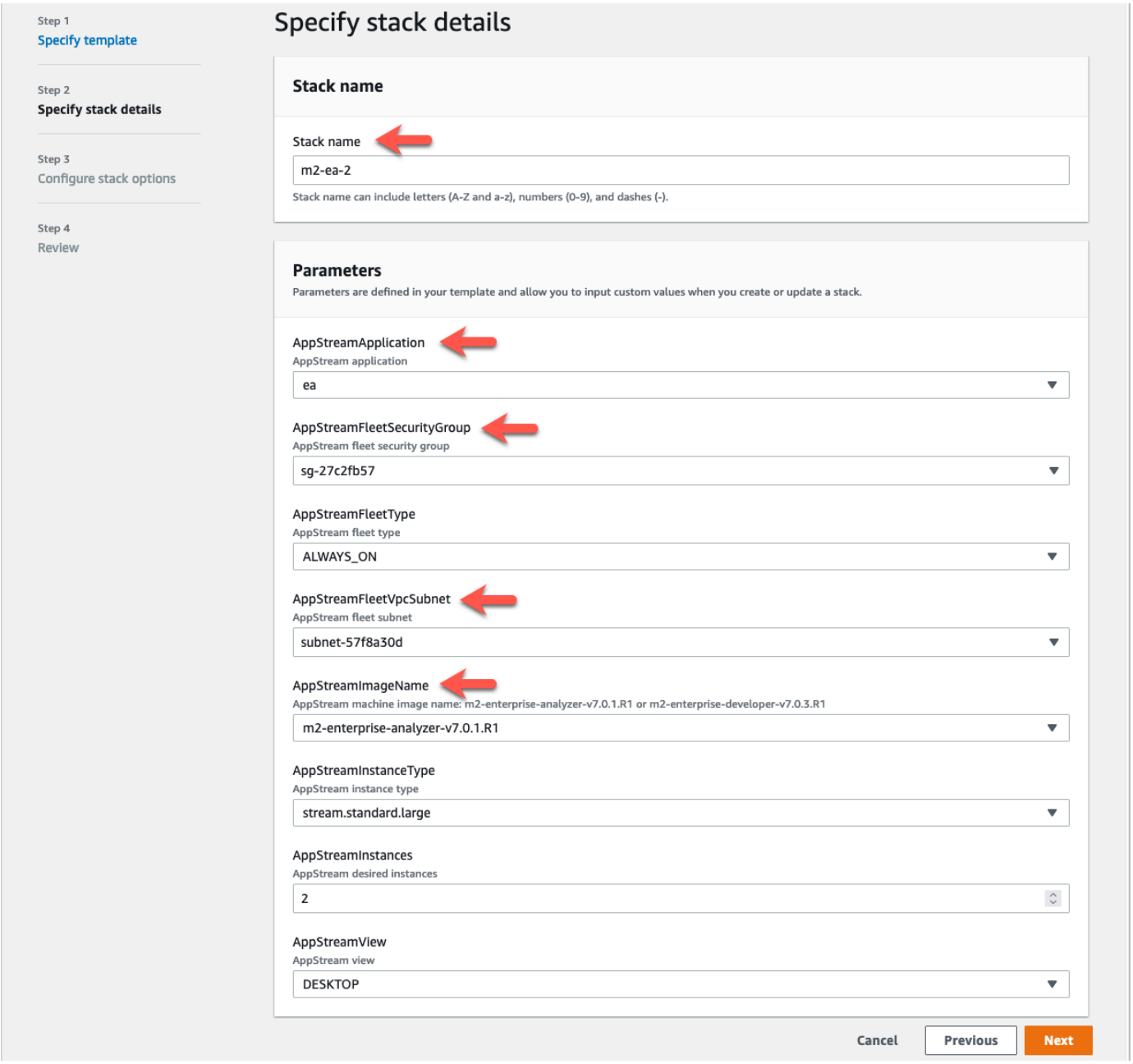
> **Note**
>
> AWS Bei der Mainframe-Modernisierung fällt für die Nutzung von Enterprise Analyzer und Enterprise Developer eine zusätzliche Gebühr zum Standard AppStream 2.0-Preis an. Weitere Informationen finden Sie unter Preise für die AWS Mainframe-Modernisierung.

1. Laden Sie bei Bedarf die Vorlage cfn-m2- appstream-fleet-ea-ed .yml herunter.
2. Öffnen Sie die AWS CloudFormation Konsole und wählen Sie Create Stack and with new resources (Standard).
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.




7. Geben Sie unter Stack-Details angeben die folgenden Informationen ein:

- Geben Sie im Feld Stackname einen Namen Ihrer Wahl ein. Beispiel, **`m2-ea`**.
- In AppStreamApplication, wähle ea aus.
- Wählen Sie in AppStreamFleetSecurityGroupdie Standardsicherheitsgruppe Ihrer Standard-VPC aus.
- Wählen Sie AppStreamFleetVpcSubnetunter ein Subnetz innerhalb Ihrer Standard-VPC aus.
- Wählen Sie AppStreamImageNameunter das Bild aus, das mit beginnt. `m2-enterprise-analyzer` Dieses Bild enthält die derzeit unterstützte Version des Rocket Enterprise Analyzer-Tools.
- Akzeptieren Sie die Standardwerte für die anderen Felder und wählen Sie dann Weiter.




8. Akzeptieren Sie alle Standardwerte und wählen Sie dann erneut Weiter.
9. Stellen Sie bei der Überprüfung sicher, dass alle Parameter Ihren Vorstellungen entsprechen.
10. 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

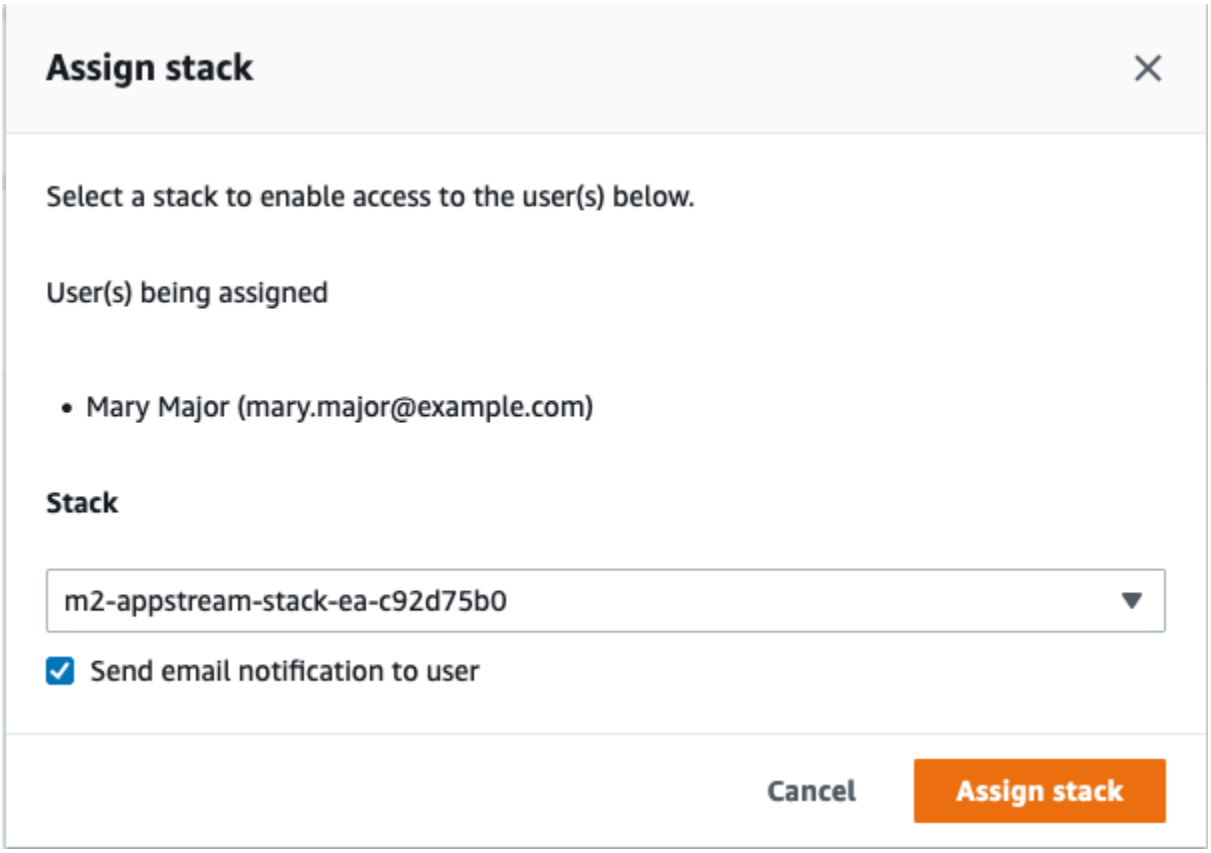
Während Sie darauf warten, AWS CloudFormation dass die Erstellung des Stacks abgeschlossen ist, können Sie in AppStream 2.0 einen oder mehrere Benutzer erstellen. Diese Benutzer werden Enterprise Analyzer in AppStream 2.0 verwenden. Sie müssen für jeden Benutzer eine E-Mail-Adresse angeben und sicherstellen, 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 beginnt`m2-appstream-stack-ea`.
5. Wählen Sie 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 habenSchritt 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 Fleets 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 How Application Settings Persistence Works 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: Rocket 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 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: Vorlagen mit Rocket Enterprise Developer (ehemals Micro Focus Enterprise Developer) verwenden

Dieses Tutorial beschreibt, wie Sie Vorlagen und vordefinierte Projekte mit Rocket 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.

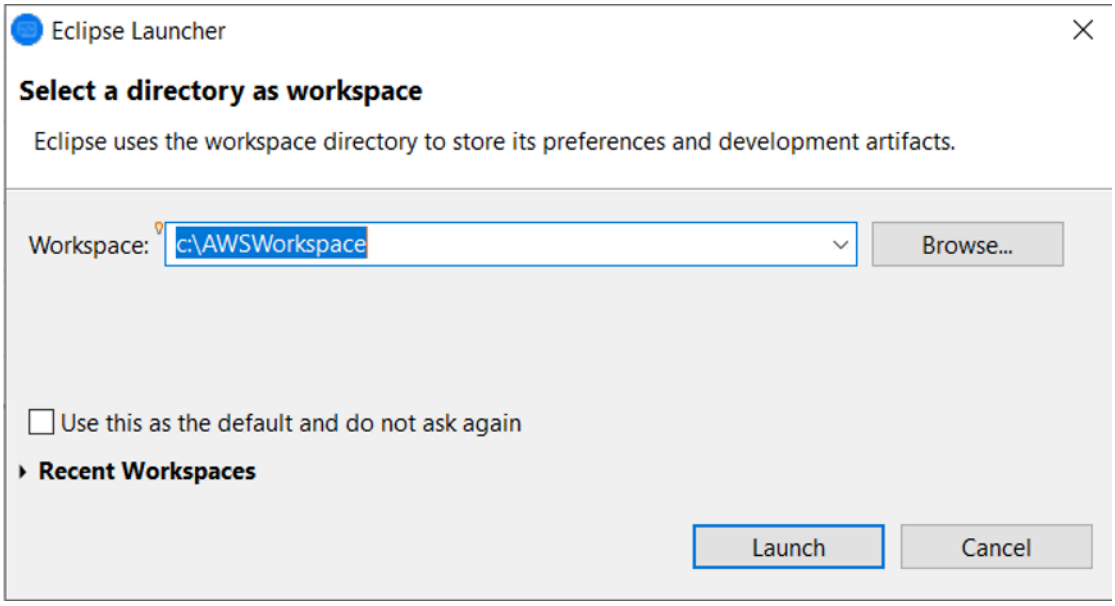
Themen

- 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

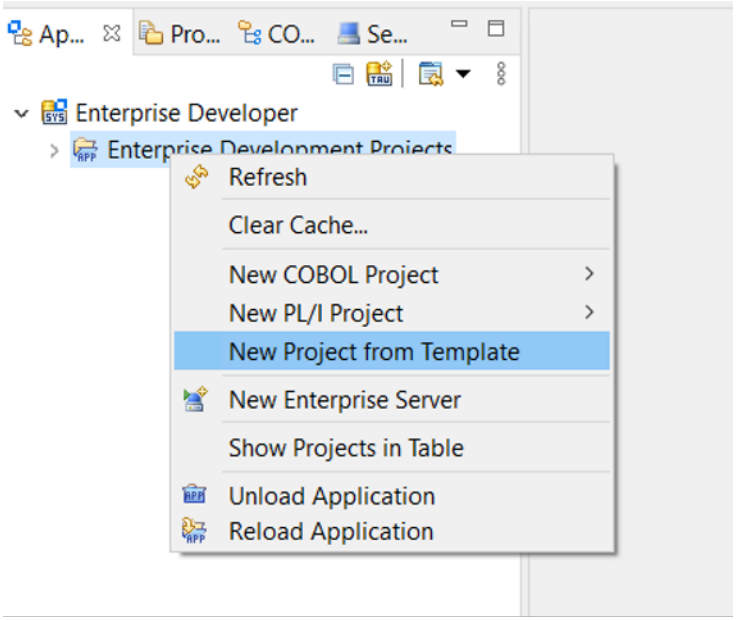
## Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält

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.

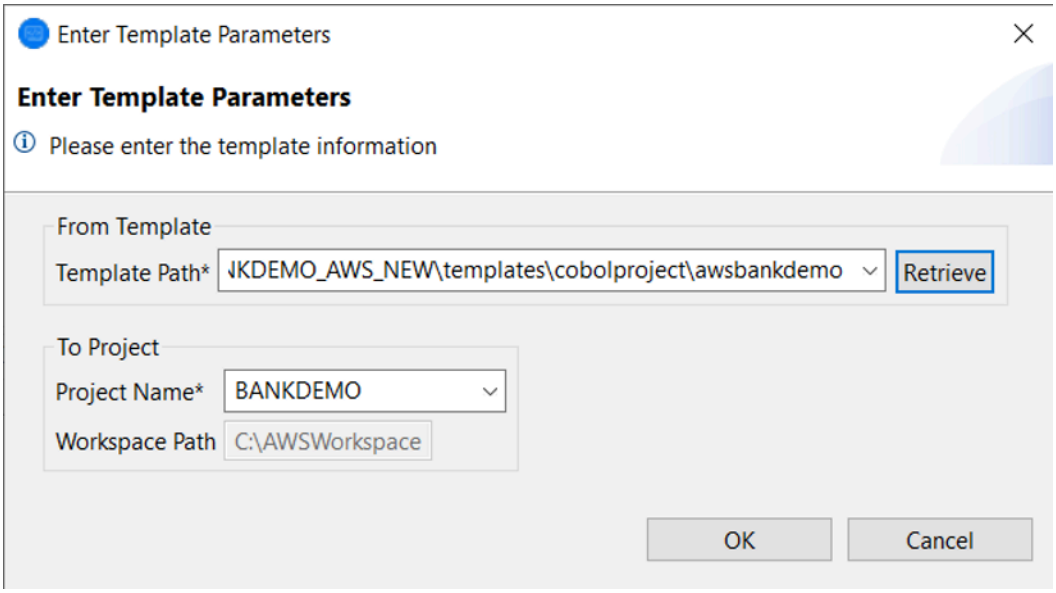



2. Wählen Sie in der Ansicht des Application Explorers in der Strukturansicht des Enterprise Development-Projekts im Kontextmenü die Option Neues Projekt aus Vorlage aus.

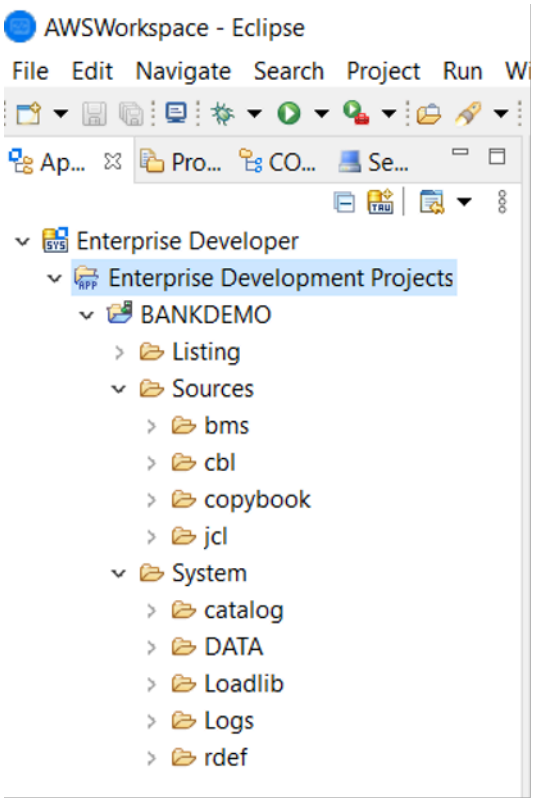



3. Geben Sie die Vorlagenparameter wie in der Abbildung gezeigt ein.

> **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.

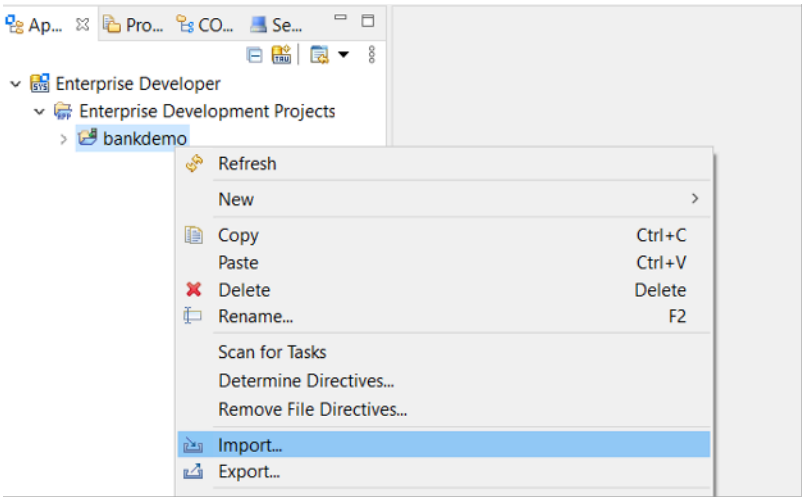
## Anwendungsfall 2 — Verwendung der COBOL-Projektvorlage ohne Quellkomponenten

Die Schritte 1 bis 3 sind identisch mit. Anwendungsfall 1 — Verwendung der COBOL-Projektvorlage, die Quellkomponenten enthält

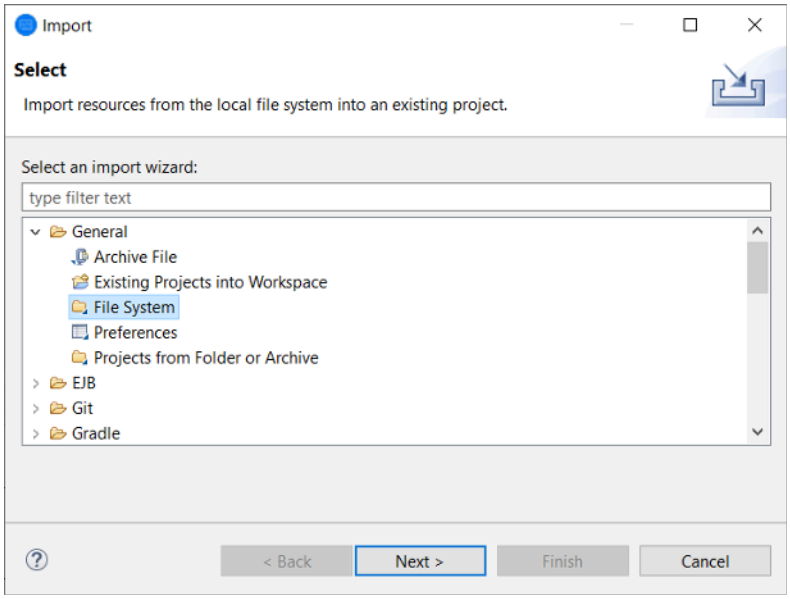
Die `System` Struktur in diesem Anwendungsfall enthält auch eine vollständige Ressourcendefinitionsdatei mit den erforderlichen Einträgen für BankDemo, den erforderlichen Katalog mit hinzugefügten Einträgen und den entsprechenden ASCII-Datendateien.

Die Quellstruktur der Vorlage enthält jedoch keine Komponenten. Sie müssen diese aus dem Quell-Repository, das Sie verwenden, in das Projekt importieren.

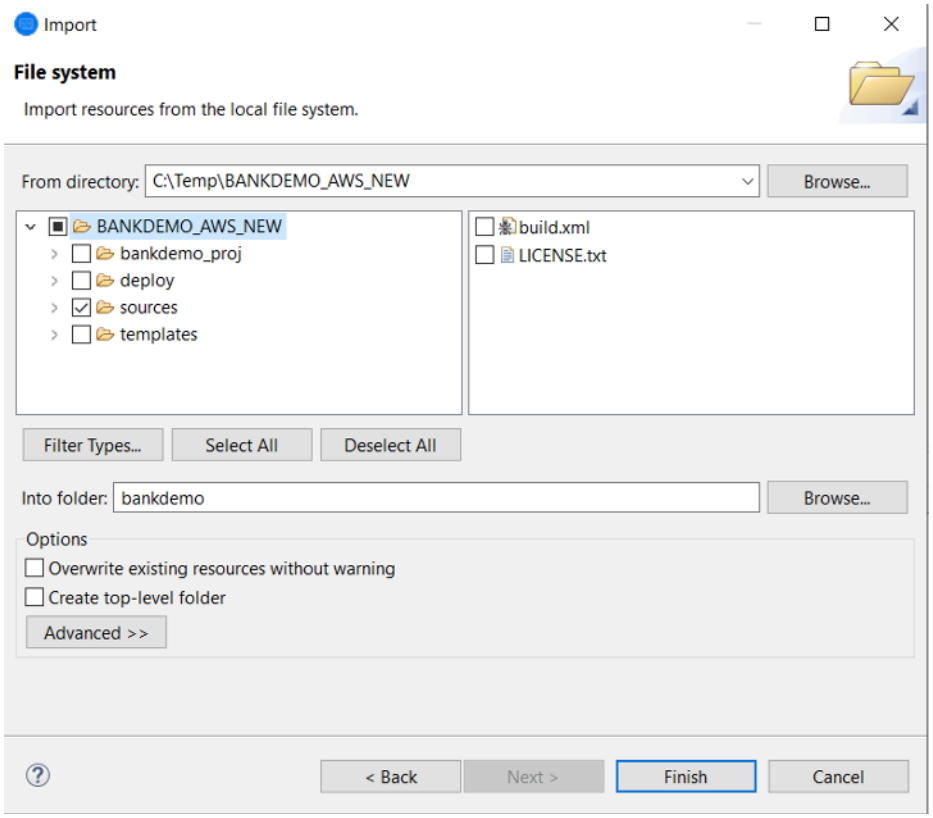
1. Wählen Sie den Projektnamen. Wählen Sie im entsprechenden Kontextmenü die Option Import.




2. Wählen Sie im daraufhin angezeigten Dialogfeld im Bereich Allgemein die Option Dateisystem und dann Weiter.



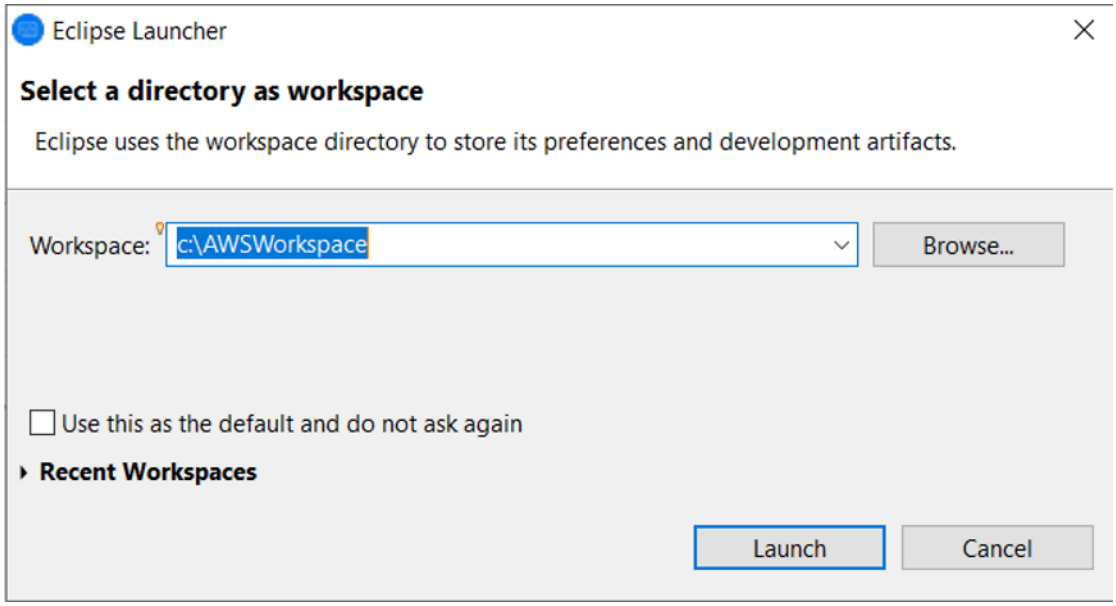

3. Füllen Sie das Feld Aus dem Verzeichnis aus, indem Sie das Dateisystem so durchsuchen, dass es auf den Repository-Ordner verweist. Wählen Sie alle Ordner aus, die Sie importieren möchten, z. B. `sources` Das `Into folder` Feld wird vorab ausgefüllt. Wählen Sie Finish (Abschließen).



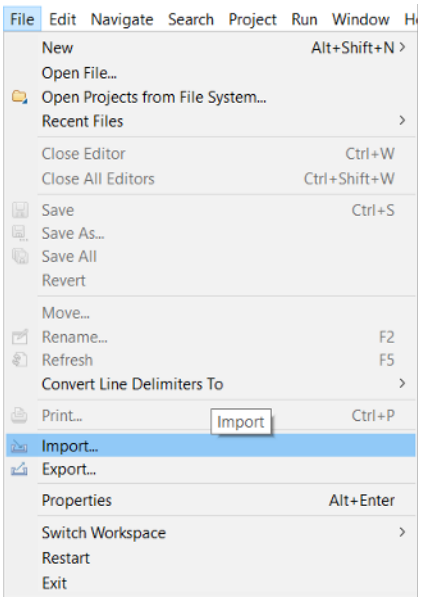

Nachdem die Quellvorlagenstruktur alle Quellelemente enthält, werden sie automatisch in Enterprise Developer erstellt.

## Anwendungsfall 3 — Verwendung des vordefinierten COBOL-Projekts, das auf die Quellordner verweist

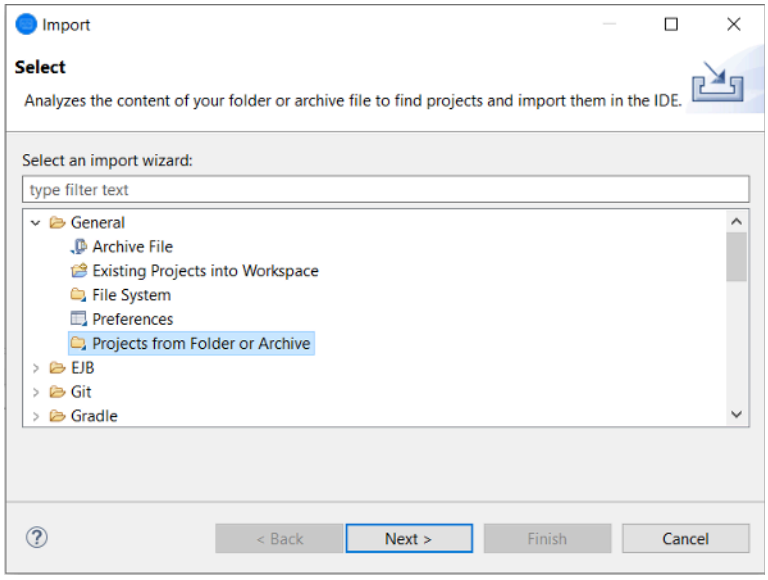
1. Starten Sie Enterprise Developer und geben Sie den ausgewählten Workspace an.



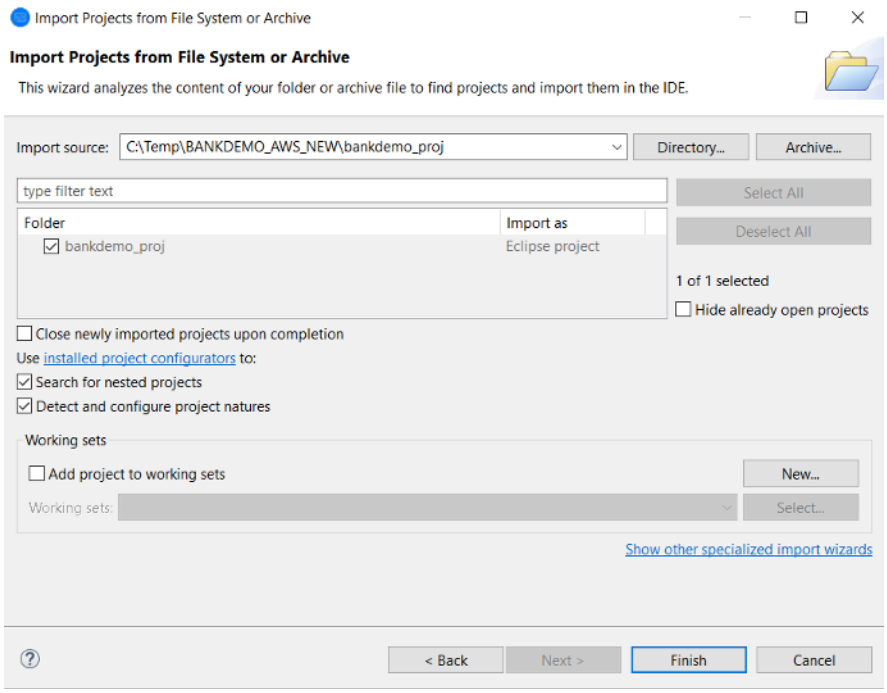

2. Wählen Sie im Menü Datei die Option Import aus.




3. Wählen Sie im daraufhin angezeigten Dialogfeld unter Allgemein die Option Projekte aus Ordner oder Archiv und dann Weiter.




4. 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.

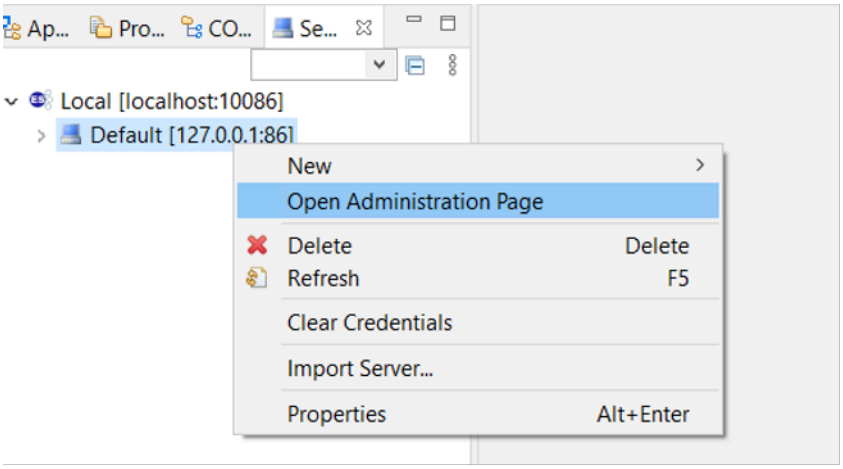



Wählen Sie Finish (Abschließen).

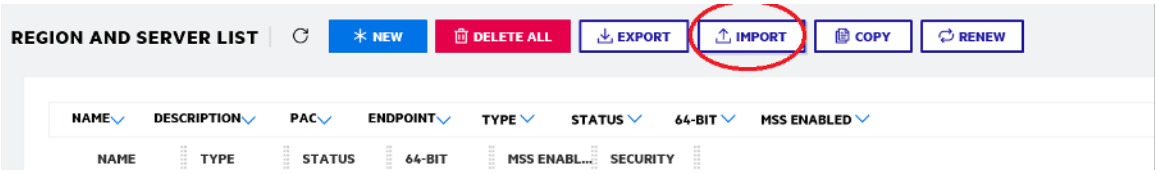
Da das Projekt durch die Links zum Quellordner aufgefüllt wird, wird der Code automatisch erstellt.

## Verwenden der JSON-Vorlage für die Regionsdefinition

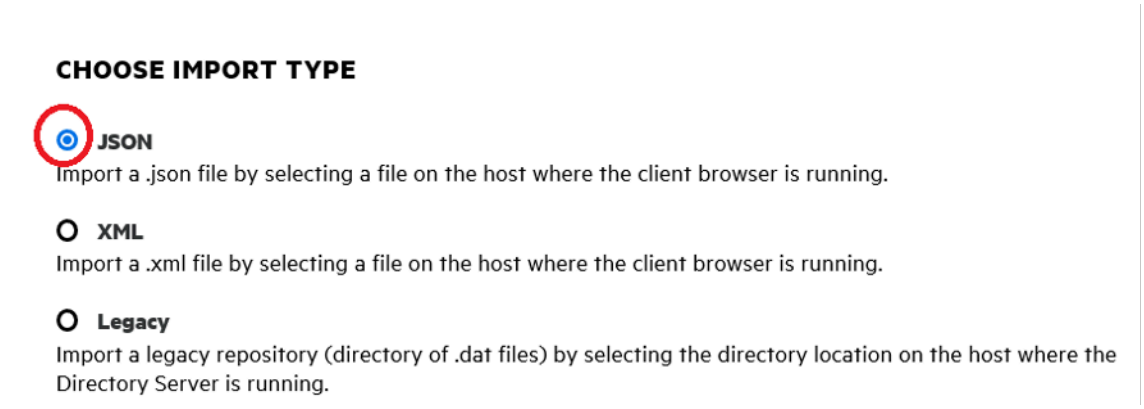
1. Wechseln Sie zur Server Explorer-Ansicht. Wählen Sie im entsprechenden Kontextmenü die Option Administrationsseite öffnen, wodurch der Standardbrowser gestartet wird.



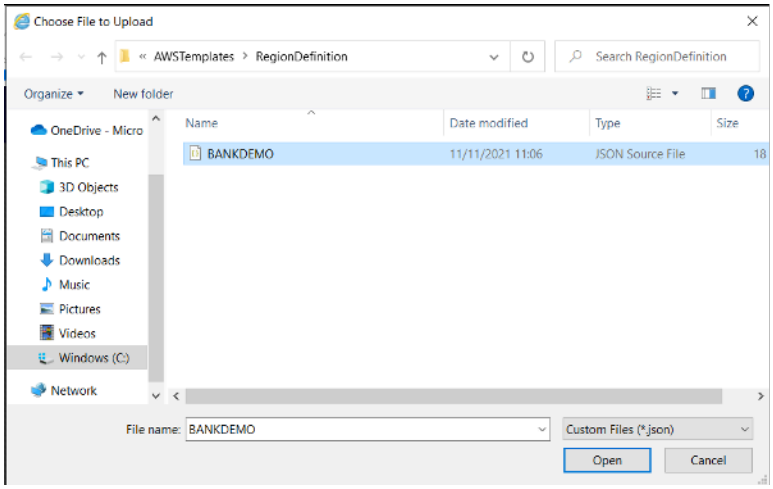

2. Wählen Sie im daraufhin angezeigten Fenster Enterprise Server Common Web Administration (ESCWA) die Option Import aus.



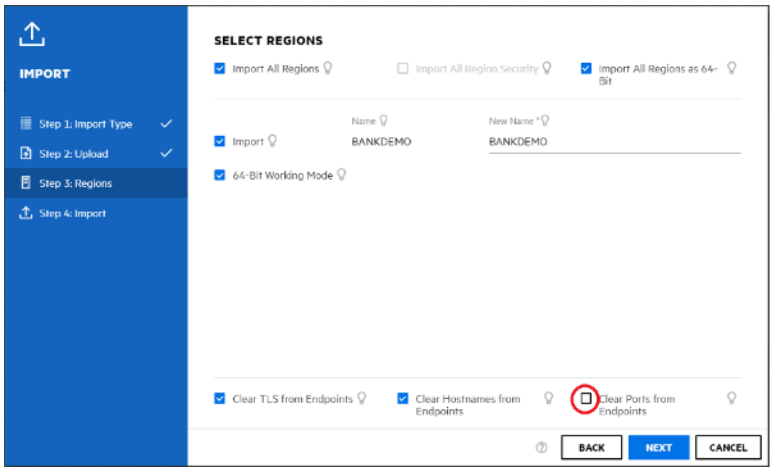

3. Wählen Sie den JSON-Importtyp und dann Weiter.




4. Laden Sie die bereitgestellte `BANKDEMO.JSON` Datei hoch.

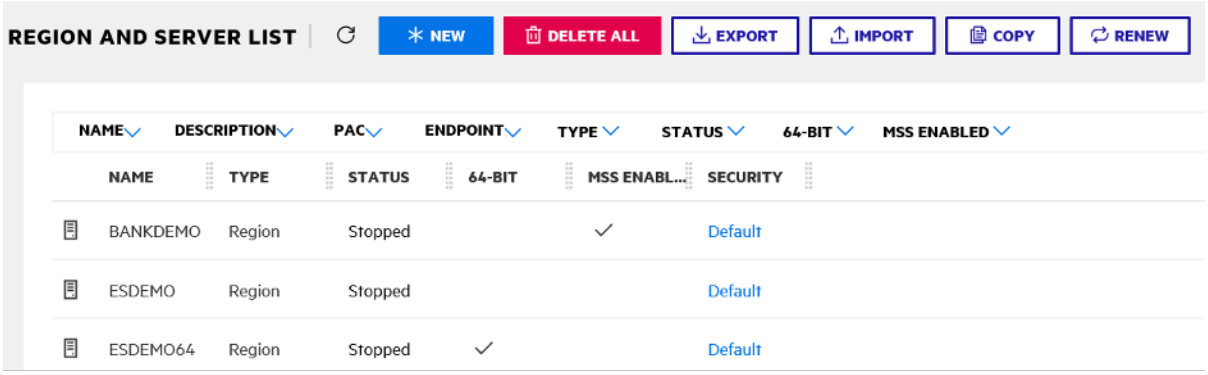



Sobald Sie die Auswahl getroffen haben, wählen Sie Weiter.

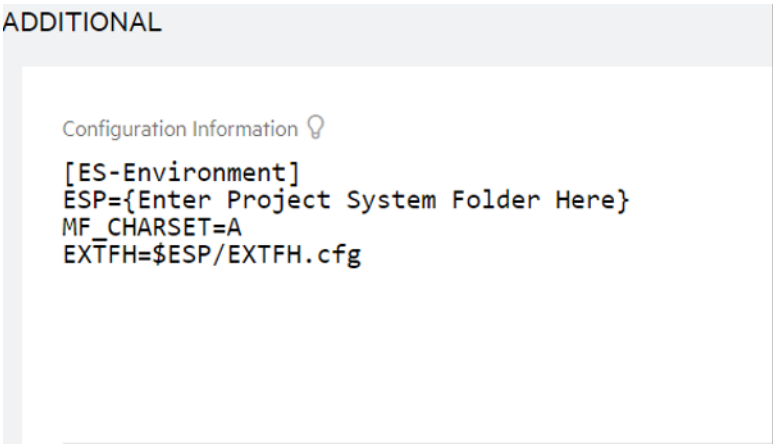



Stellen Sie sicher, dass im Bereich „Regionen auswählen“ die Option „Anschlüsse von Endpunkten löschen“ nicht aktiviert ist, und wählen Sie dann in den Bedienfeldern „Weiter“, bis der Bereich „Import durchführen“ angezeigt wird. Wählen Sie dann im linken Navigationsbereich Import aus.

Klicken Sie abschließend auf Fertig stellen. Die BANKDEMO-Region wird dann der Serverliste hinzugefügt.

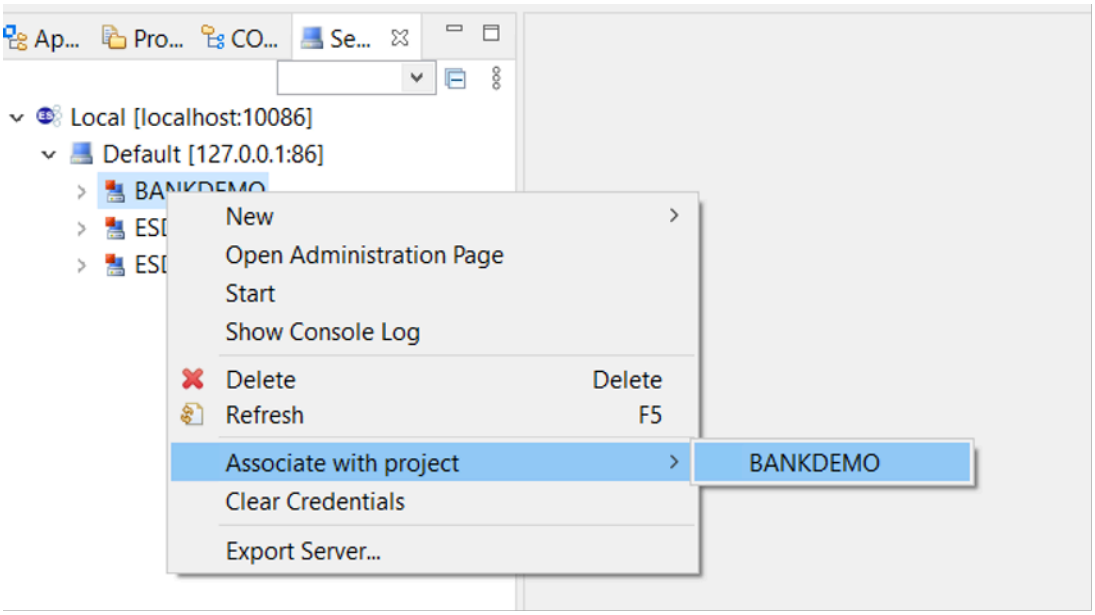



5. Navigieren Sie zu den Allgemeinen Eigenschaften für die Region BANKDEMO.
6. Scrollen Sie zum Abschnitt Konfiguration.
7. Die ESP-Umgebungsvariable muss auf den `System` Ordner gesetzt werden, der für das in den vorherigen Schritten erstellte Eclipse-Projekt relevant ist. Das sollte sein`workspacefolder/projectname/System`.




8. Klicken Sie auf Apply (Anwenden).

   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 mit einer vollständigen, funktionierenden Version von einsatzbereit 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: Enterprise Analyzer auf AppStream 2.0 einrichten

Dieses Tutorial beschreibt, wie Sie Rocket Enterprise Analyzer (früher 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 Rocket Enterprise Analyzer und Rocket 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 - .yml eingerichtet haben. appstream-fleet-ea-ed Weitere Informationen finden Sie unter Tutorial: AppStream 2.0 für die Verwendung mit Rocket Enterprise Analyzer und Rocket 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 Website von Rocket Software (ehemals Micro Focus).

## Inhalt des Bildes

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 Programmgesteuerter Zugriff in der.Allgemeine 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-rclone.cmd` die folgenden Änderungen vor:

   - Wechseln Sie `amzn-s3-demo-bucket` zu Ihrem Amazon S3 S3-Bucket-Namen. Beispiel, `m2-s3-mybucket`.
   - Wechseln Sie `your-s3-folder-key` zu Ihrem Amazon S3 S3-Bucket-Key. Beispiel, `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. Beispiel, `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:amzn-s3-demo-bucket/your-s3-
   folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:
   \PhotonUser\My Files\Home Folder\m2-rclone.conf"
   goto :loop
   ```

4. 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 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-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: 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 dann 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. Loggen Sie sich mit Ihrer E-Mail-Adresse und Ihrem permanenten Passwort ein.

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.

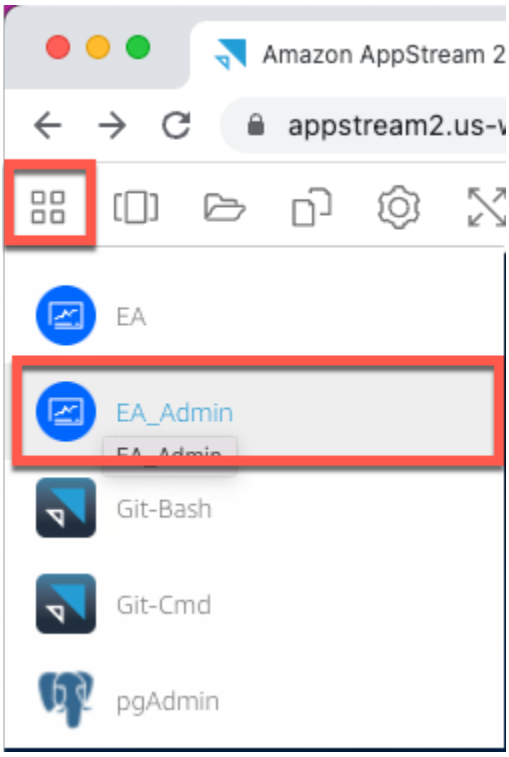
## Problembehandlung bei der Workspace-Verbindung

Wenn Sie versuchen, erneut eine Verbindung zu Ihrem Enterprise Analyzer-Workspace herzustellen, wird möglicherweise ein Fehler wie der folgende angezeigt:

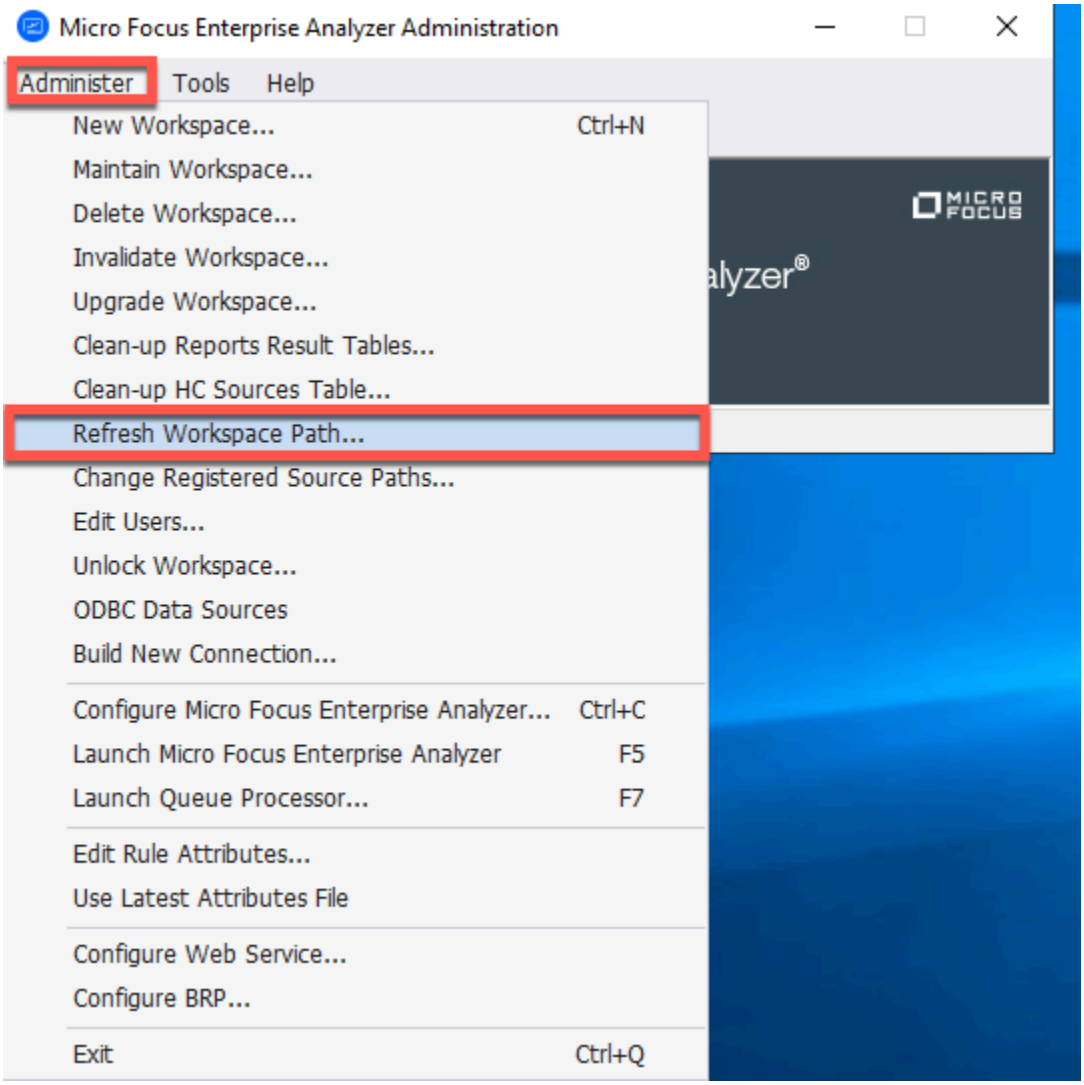
```
Cannot access the workspace directory D:\PhotonUser\My Files\Home Folder\EA_BankDemo.
 The workspace has been created on a non-shared disk of the EC2AMAZ-E6LC33H computer.
 Would you like to correct the workspace directory location?
```

Um dieses Problem zu beheben, wählen Sie OK, um die Meldung zu löschen, und führen Sie dann die folgenden Schritte aus.

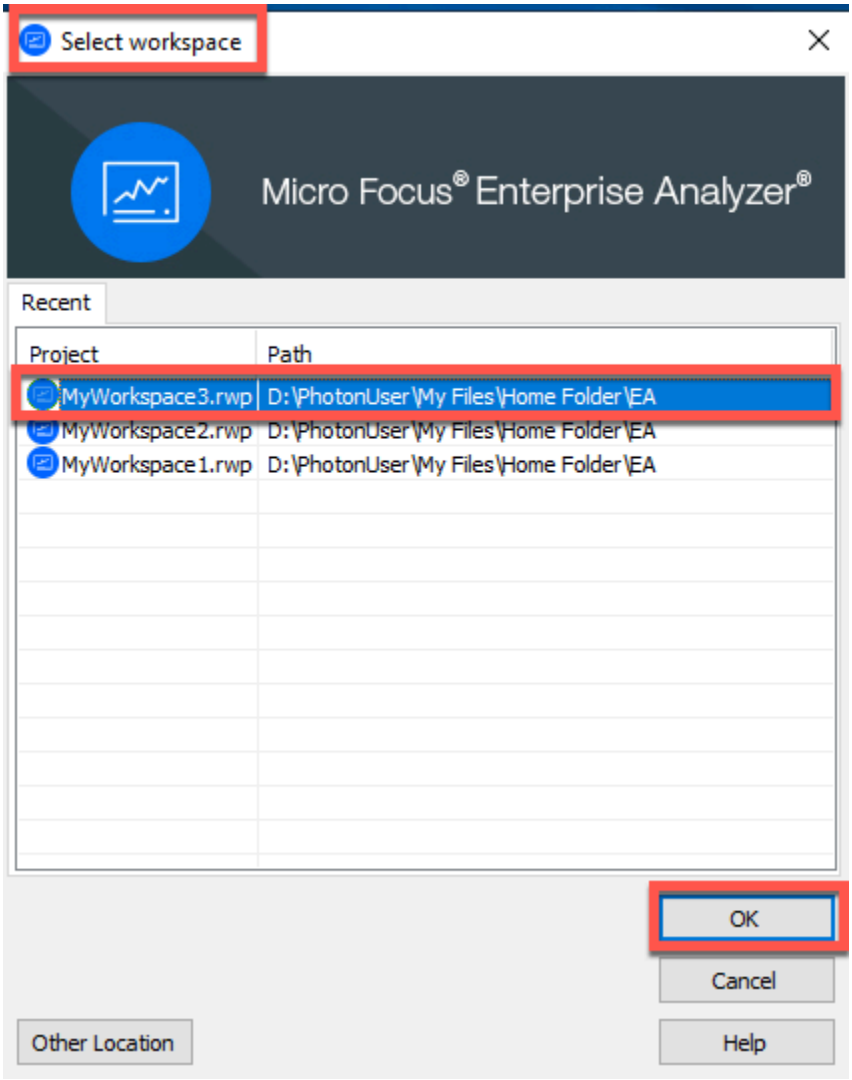
1. Wählen Sie in AppStream 2.0 in der Werkzeugleiste das Symbol „Anwendung starten“ und anschließend „EA_Admin“, um das Enterprise Analyzer-Administrationstool zu starten.



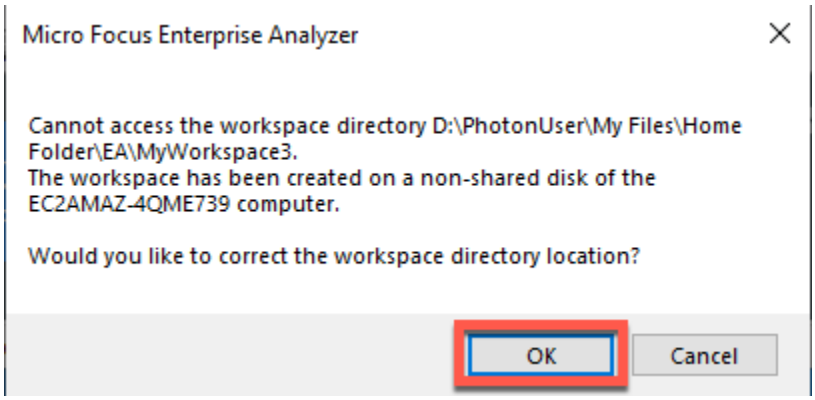

2. Wählen Sie im Menü „Verwalten“ die Option Workspace-Pfad aktualisieren... .



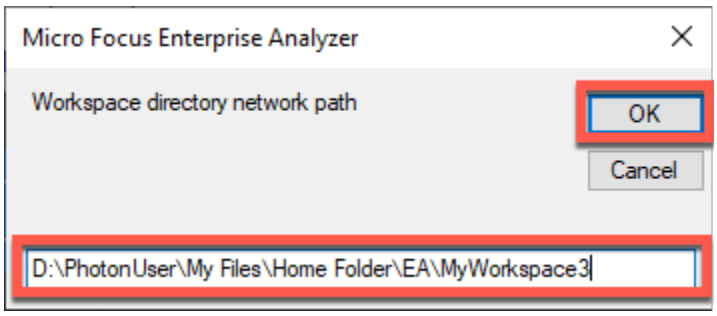

3. Wählen Sie unter Arbeitsbereich auswählen den gewünschten Arbeitsbereich aus und klicken Sie dann auf OK.



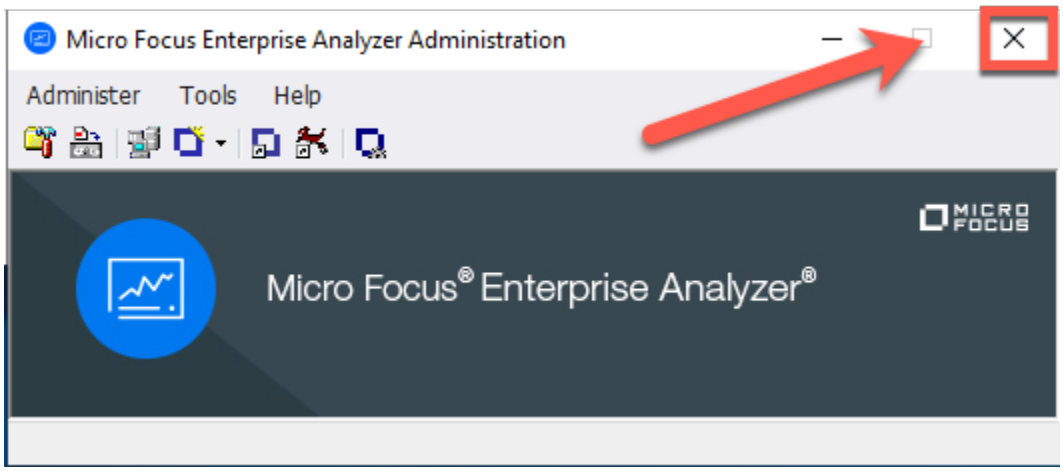

4. Wählen Sie OK, um die Fehlermeldung zu bestätigen.



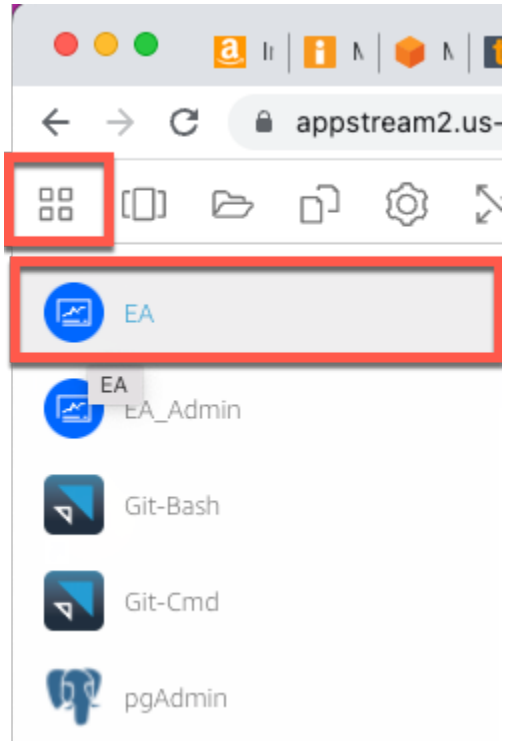

5. Geben Sie unter Netzwerkpfad für das Workspace-Verzeichnis den richtigen Pfad zu Ihrem Workspace ein, `D:\PhotonUser\My Files\Home Folder\EA\MyWorkspace3` z. B.




6. Schließen Sie das Micro Focus Enterprise Analyzer Administration Tool.




7. 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: Rocket Enterprise Developer auf AppStream 2.0 einrichten

Dieses Tutorial beschreibt, wie Sie Rocket Enterprise Developer (ehemals 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 Rocket Enterprise Analyzer und Rocket 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 - .yaml einrichten. appstream-fleet-ea-ed Weitere Informationen finden Sie unter Tutorial: AppStream 2.0 für die Verwendung mit Rocket Enterprise Analyzer und Rocket 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 Website von Rocket Software (ehemals Micro Focus).

## Inhalt des Bildes

Zusätzlich zu Enterprise Developer selbst enthält das Image Rumba (einen TN327 0-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 von entsprechen CI/CD siehe oben genannte Pipeline, 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 IAM-Autorisierungen für CodeCommit finden Sie in der CodeCommit Berechtigungsreferenz: Der Administrator kann unterschiedliche IAM-Richtlinien für verschiedene Rollen definieren, wobei die 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. Connect zur AWS Konsole unter her https://console.aws.amazon.com/iam/.
    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 für die Verbindung mit CodeCommit.

2. Starten Sie eine Sitzung mit AppStream 2.0, basierend auf 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 `amzn-s3-demo-bucket` zu Ihrem Amazon S3 S3-Bucket-Namen. Beispiel, `m2-s3-mybucket`.
- Wechseln Sie `your-s3-folder-key` zu Ihrem Amazon S3 S3-Bucket-Key. Beispiel, `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. Beispiel, `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:amzn-s3-demo-bucket/your-s3-
folder-key "D:\PhotonUser\My Files\Home Folder\your-local-folder-path" --config "D:
\PhotonUser\My Files\Home Folder\m2-rclone.conf"
goto :loop
```

4. 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 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-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 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 Repositorys, 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`Rclone`, 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.

## Verfügbare Batch-Dienstprogramme in AWS Mainframe Modernization

Mainframe-Anwendungen verwenden häufig Batch-Hilfsprogramme, um bestimmte Funktionen auszuführen, z. B. das Sortieren von Daten, das Übertragen von Dateien über FTP, das Laden von Daten in Datenbanken DB2, das Entladen von Daten aus Datenbanken usw.

Wenn Sie Ihre Anwendungen auf AWS Mainframe Modernization migrieren, benötigen Sie funktionsgleiche Ersatzprogramme, 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 Ausführung von Befehlen, Skripten und Systemaufrufen von Rocket Software (ehemals Micro Focus) direkt 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 Rocket Enterprise Developer (ED) und Rocket Software (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 Ihr Mainframe-FTP JCLs zur Verwendung von SFTP zu konvertieren, wechseln `PGM=FTP` Sie zu. `PGM=M2SFTP`

Themen


- Unterstützte Plattformen
- Installieren von Abhängigkeiten
- Konfigurieren Sie M2SFTP für Mainframe Modernization Managed AWS
- M2SFTP für AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich 2.0) AppStream
- Beispiel JCLs
- Befehlsreferenz für den Putty SFTP (PSFTP) -Client
- Nächste Schritte


## Unterstützte Plattformen

Sie können M2SFTP auf jeder der folgenden Plattformen verwenden:

- AWS Von Rocket Software (ehemals Micro Focus) verwaltete Mainframe-Modernisierung
- Rocket Software Runtime (bei Amazon EC2)
- Alle Varianten der Produkte Rocket Software Enterprise Developer (ED) und Rocket Software 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 Rocket Enterprise Server-Umgebungsvariablen für MFFTP ein. Hier sind ein paar 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 JCLs

## M2SFTP für AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich 2.0) AppStream

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon laufen EC2, 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 Rocket Enterprise Server-Umgebungsvariablen für MFFTP ein. Hier sind ein paar 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 JCLs

## Beispiel JCLs

Um die Installation zu testen, können Sie eine der folgenden JCL-Beispieldateien verwenden.

M2 .jcl SFTP1

Diese JCL zeigt, wie M2SFTP aufgerufen wird, 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
/*
//*
//
```

### M2 SFTP2 .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 |

| Befehl | Beschreibung |
| --- | --- |
| 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 |
| mget | Laden Sie mehrere Dateien gleichzeitig herunter |
| mkdir | Erstellen Sie Verzeichnisse auf dem Remoteser ver |
| mput | Laden Sie mehrere Dateien gleichzeitig hoch |
| mv | Datei (en) auf dem Remoteserver verschieben oder umbenennen |
| geöffnet | Stellen Sie eine Connect zu einem Host her |
| 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 |

| Befehl | Beschreibung |
| --- | --- |
| 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 SFTPAWS 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 Rocket Software (ehemals 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-Modernisierung (Managed)
- M2WAIT for AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich AppStream 2.0)
- Beispiel JCL

## Unterstützte Plattformen

Sie können M2WAIT auf jeder der folgenden Plattformen verwenden:

- AWS Von Rocket Software (ehemals Micro Focus) verwaltete Mainframe-Modernisierung
- Rocket Software Runtime (bei Amazon EC2)
- Alle Varianten der Produkte Rocket Software Enterprise Developer (ED) und Rocket Software Enterprise Server (ES).

## Konfigurieren Sie M2WAIT für AWS Mainframe-Modernisierung (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 for AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich AppStream 2.0)

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon laufen EC2, 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 TXT2 PDF (z/

OS Freeware). Wir haben es so modifiziert, dass es unter der Laufzeitumgebung AWS Mainframe Modernization Rocket Software (ehemals Micro Focus) läuft.

Themen


- Unterstützte Plattformen
- TXT2PDF für AWS Mainframe Modernization Managed konfigurieren
- TXT2PDF für AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich AppStream 2.0)
- Beispiel JCL
- Modifikationen
- Referenzen


## Unterstützte Plattformen

Sie können TXT2 PDF auf jeder der folgenden Plattformen verwenden:

- AWS Mainframe-Modernisierung, Rocket Software Managed
- Rocket Software Runtime (bei Amazon EC2)
- Alle Varianten der Produkte Rocket Enterprise Developer (ED) und Rocket Enterprise Server (ES).

## TXT2PDF für AWS Mainframe Modernization Managed konfigurieren

Wenn Ihre migrierten Anwendungen auf AWS Mainframe Modernization Managed ausgeführt werden, konfigurieren Sie TXT2 PDF wie folgt.

- Erstellen Sie eine REXX EXEC-Bibliothek mit dem Namen. `AWS.M2.REXX.EXEC` Laden Sie diese REXX-Module herunter und kopieren Sie sie in die Bibliothek.
  - `TXT2PDF.rex`- PDF TXT2 z/OS Freeware (modifiziert)
  - `TXT2PDFD.rex`- PDF TXT2 z/OS Freeware (unverändert)
  - `TXT2PDFX.rex`- PDF TXT2 z/OS Freeware (modifiziert)
  - `M2GETOS.rex`- Um den Betriebssystemtyp zu überprüfen (Windows oder Linux)

Informationen zum Testen Ihrer Konfiguration finden Sie unterBeispiel JCL.

## TXT2PDF für AWS Mainframe Modernization Runtime auf Amazon konfigurieren EC2 (einschließlich AppStream 2.0)

Wenn Ihre migrierten Anwendungen auf der AWS Mainframe Modernization Runtime auf Amazon laufen EC2, konfigurieren Sie TXT2 PDF wie folgt.

1. Stellen Sie die Rocket Software-Umgebungsvariable `MFREXX_CHARSET` auf den entsprechenden Wert ein, z. B. „A“ für ASCII-Daten.

   > **Important**
   >
   > Die Eingabe eines falschen Werts könnte zu Problemen bei der Datenkonvertierung (von EBCDIC nach ASCII) führen, wodurch das resultierende PDF unlesbar oder funktionsunfähig wird. `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`- PDF TXT2 z/OS Freeware (modifiziert)
   - `TXT2PDFD.rex`- PDF TXT2 z/OS Freeware (unverändert)
   - `TXT2PDFX.rex`- PDF TXT2 z/OS Freeware (modifiziert)
   - `M2GETOS.rex`- Um den Betriebssystemtyp zu überprüfen (Windows oder Linux)

Informationen zum Testen Ihrer Konfiguration finden Sie unterBeispiel 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 TXT2 PDF-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

Dieses JCL-Beispiel verwendet einen DSN-Namen für die PDF-Konvertierung. TXT2

```
//TXT2PDF2 JOB 'TXT2PDF2',CLASS=A,MSGCLASS=X,TIME=1440
//*
//* Copyright Amazon.com, Inc. or its affiliates.*
//* All Rights Reserved.*
//*
//*--------------------------------------------------------------------**
//* PRE DELETE*
//*--------------------------------------------------------------------**
//*
//PREDEL  EXEC PGM=IEFBR14
//*
//DD01     DD DSN=AWS.M2.TXT2PDF2.PDF.VB,
//            DISP=(MOD,DELETE,DELETE)
//*
//DD02     DD DSN=AWS.M2.TXT2PDF2.PDF,
```

```
//            DISP=(MOD,DELETE,DELETE)
//*
//*--------------------------------------------------------------------**
//* CALL TXT2PDF TO CONVERT FROM TEXT TO PDF (VB)*
//*--------------------------------------------------------------------**
//*
//STEP01 EXEC PGM=IKJEFT1B
//*
//SYSEXEC  DD DISP=SHR,DSN=AWS.M2.REXX.EXEC
//*
//INDD     DD *
1THIS IS THE FIRST LINE ON THE PAGE 1
0THIS IS THE THIRD LINE ON THE PAGE 1
-THIS IS THE   6TH LINE ON THE PAGE 1
THIS IS THE   7TH LINE ON THE PAGE 1
+_____________________________________ - OVERSTRIKE 7TH LINE
1THIS IS THE FIRST LINE ON THE PAGE 2
0THIS IS THE THIRD LINE ON THE PAGE 2
-THIS IS THE   6TH LINE ON THE PAGE 2
THIS IS THE   7TH LINE ON THE PAGE 2
+_____________________________________ - OVERSTRIKE 7TH LINE
/*
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD DDNAME=SYSIN
//*
//SYSIN    DD *
%TXT2PDF BROWSE Y IN DD:INDD +
OUT 'AWS.M2.TXT2PDF2.PDF.VB' +
CC YES
/*
//*
//*--------------------------------------------------------------------**
//* CONVERT PDF (VB) TO PDF (LSEQ - BYTE STREAM)*
//*--------------------------------------------------------------------**
//*
//STEP02 EXEC PGM=VB2LSEQ
//*
//INFILE   DD DSN=AWS.M2.TXT2PDF2.PDF.VB,DISP=SHR
//*
//OUTFILE  DD DSN=AWS.M2.TXT2PDF2.PDF,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(LRECL=256,DSORG=PS,RECFM=LSEQ,BLKSIZE=0)
//*
```

```
//SYSOUT   DD SYSOUT=*
//*
//
```

### Modifikationen

Damit das TXT2 PDF-Programm auf der AWS Mainframe Modernization Rocket Software-Laufzeitumgebung ausgeführt werden kann, haben wir die folgenden Änderungen vorgenommen:

- Änderungen am Quellcode, um die Kompatibilität mit der REXX-Laufzeit von Rocket Software sicherzustellen
- Änderungen, um sicherzustellen, dass das Programm sowohl auf Windows- als auch auf Linux-Betriebssystemen ausgeführt werden kann
- Änderungen zur Unterstützung sowohl der EBCDIC- als auch der ASCII-Laufzeit

### Referenzen

TXT2PDF-Referenzen und Quellcode:

- Text-zu-PDF-Konverter
- z/OS Kostenlose TCP/IP- und 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 JCLs

## Unterstützte Plattformen

Sie können M2DFUTIL auf jeder der folgenden Plattformen verwenden:

- Rocket Software (ehemals Micro Focus) ES unter Windows (64 Bit und 32 Bit)
- Rocket Software 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, CREDT, 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. Verarbeitete Dateien 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 Massenlöschen 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 DSNAME 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 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 Befehle, Skripte und Systemaufrufe von Rocket Software (ehemals Micro Focus) 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 JCLs

## Unterstützte Plattformen

Sie können M2RUNCMD auf den folgenden Plattformen verwenden:

- Rocket Software Runtime (bei Amazon EC2)
- Alle Varianten der Produkte Rocket Software Enterprise Developer (ED) und Rocket Software 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 laufen EC2, 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 JCLs

Um die Installation zu testen, können Sie eines der folgenden Beispiele verwenden JCLs.

RUNSCRL1.jcl

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 Rocket Software- und Systembefehle auszuführen.

```
//RUNSCRL1 JOB 'RUN SCRIPT',CLASS=A,MSGCLASS=X,TIME=1440
//*
//*
//*--------------------------------------------------------------------*
//*  CREATE SCRIPT (LINUX)
//*--------------------------------------------------------------------*
//*
//STEP0010 EXEC PGM=IEBGENER
//*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//SYSUT1   DD *
#!/bin/bash

set -x

## ECHO PATH ENVIRONMNET VARIABLE
echo $PATH

## CLOSE/DISABLE VSAM FILE
casfile -r$ES_SERVER -oc  -ed -dACCTFIL

## OPEN/ENABLE VSAM FILE
casfile -r$ES_SERVER -ooi -ee -dACCTFIL

exit $?
/*
//SYSUT2   DD DSN=&&TEMP,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=LSEQ,LRECL=300,DSORG=PS,BLKSIZE=0)
```

```
//*MFE: %PCDSN='/tmp/TEST_SCRIPT.sh'
//*
//*--------------------------------------------------------------------*
//*   RUN SCRIPT (LINUX)                                               *
//*--------------------------------------------------------------------*
//*
//STEP0020 EXEC PGM=RUNCMD
//*
//SYSOUT  DD  SYSOUT=*
//*
//SYSIN   DD *
*RUN SCRIPT
 sh /tmp/TEST_SCRIPT.sh
/*
//
```

### SYSOUT

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  ************
```

RUNCMDL1.jcl

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
/*
//
```

# 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. Es vereinfacht den Prozess der Übertragung von Datensätzen von Ihrem Mainframe zum. AWS Cloud Zu den wichtigsten Funktionen gehören: Erkennung von Mainframe-Quelldatensätzen und Artefakten sowie Skalierbarkeit und Effizienz für schnellere Datenübertragungen zu Amazon S3. File Transfer unterstützt verschiedene Mainframe-Datensatztypen wie sequentiell, PDS, GDS, GDG und VSAM KSDS. Der Service überträgt die Datensätze in einen Amazon S3 S3-Zwischenbucket, konvertiert sie in die angegebene Ziel-Codepage und verschiebt sie dann in den gewünschten Ziel-S3-Bucket.

Themen


- Was ist AWS Mainframe Modernization File Transfer?
- Installieren Sie einen File Transfer Agent
- Einen File Transfer Agent konfigurieren
- Datenübertragungsendpunkte für die Dateiübertragung erstellen
- Übertragungsaufgaben in File Transfer erstellen
- Tutorial: Erste Schritte mit AWS Mainframe Modernization File Transfer
- Unterstützte Quell- und Zielkodierungen in AWS Mainframe Modernization File Transfer


## 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

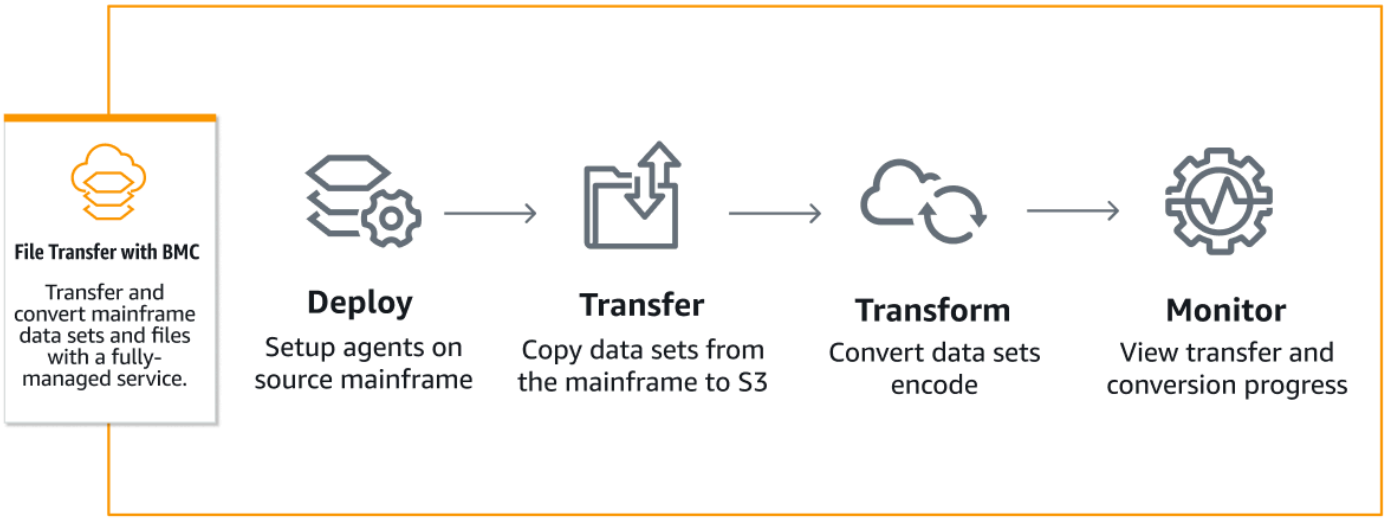
# Vorteile von AWS Mainframe Modernization File Transfer

Mit AWS Mainframe Modernization File Transfer können Sie Datensätze vom Mainframe nach Amazon S3 übertragen. Zu den Vorteilen gehören:

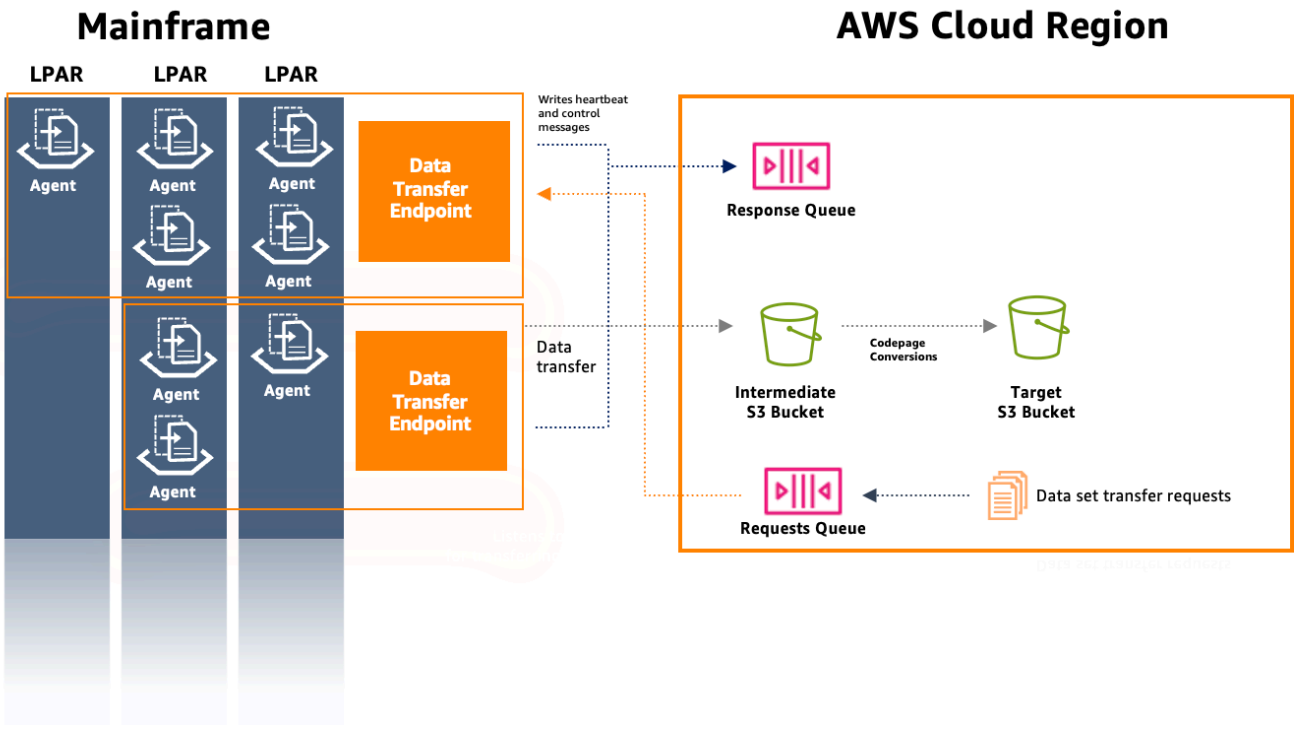
- Entdeckung von Mainframe-Quelldatensätzen und Artefakten
- Automatisierte Übertragungen und Konvertierung von Datensätzen
- Skalierbarkeit, Effizienz und Geschwindigkeit für schnellere Datenübertragungen an AWS

# 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

Sie können dieses Dokument als step-by-step Leitfaden für die Installation eines Agenten auf dem Quell-Mainframe verwenden.

> **Note**
>
> Dieses Handbuch richtet sich nur an Mainframe-Systemprogrammierer.

Themen


- Schritt 1: Erstellen Sie einen ZfS-Datensatz für den M2-Agenten
- Schritt 2: Formatieren Sie den Datensatz als ZFS
- Schritt 3: Mounten Sie das Dateisystem
- Schritt 4: Überprüfen Sie den Mount
- Schritt 5: Geben Sie OMVS ein
- Schritt 6: Legen Sie die Umgebungsvariable für das Installationsverzeichnis des Agenten fest

- Schritt 7: Legen Sie die Umgebungsvariable für das Arbeitsverzeichnis fest
- Schritt 8: Erstellen Sie das Arbeitsverzeichnis
- Schritt 9: Kopieren Sie die Tar-Datei des Agenten und das Arbeitsverzeichnis
- Schritt 10: Beenden Sie die Agenteninstallation


## Schritt 1: Erstellen Sie einen ZfS-Datensatz für den M2-Agenten

Erstellen Sie ein ZFS für die M2-Agent-Installation mit der folgenden JCL (Job Control Language):

```
DEFINE   EXEC   PGM=IDCAMS
SYSPRINT DD     SYSOUT=A
SYSIN    DD     *
 DEFINE CLUSTER (NAME(yourhlq.M2AGENT.ZFS) -
 VOLUMES(*) -
 LINEAR CYL(1000 200))
```

## Schritt 2: Formatieren Sie den Datensatz als ZFS

Nachdem Sie den Datensatz erstellt haben, formatieren Sie ihn als ZFS-Dateisystem.

Eine Möglichkeit, dies zu tun, ist die Verwendung der folgenden JCL:

```
FORMAT  EXEC PGM=IOEAGFMT,
PARM=('-aggregate yourhlq.M2AGENT.ZFS -size 1200') ,
SYSPRINT DD  SYSOUT=*
```

Reichen Sie diesen Job ein und überprüfen Sie, ob er erfolgreich abgeschlossen wurde.

## Schritt 3: 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.

Zum Beispiel:

```
MOUNT FILESYSTEM('yourhlq.M2AGENT.ZFS') TYPE(ZFS) MODE(RDWR) MOUNTPOINT('/usr/lpp/aws/
m2-agent')
```

Sie werden diesen Einhängepunkt in Schritt 6 verwenden.

> **Note**
>
> Die Definition des Einhängepfads ist optional und Sie sollten dafür ein vorhandenes Verzeichnis verwenden.

## Schritt 4: Überprüfen Sie den Mount

Stellen Sie sicher, dass das Dateisystem korrekt gemountet ist. Verwenden Sie dazu einen `D OMVS,F` Befehl oder überprüfen Sie es im Unix System Service (USS).

## Schritt 5: Geben Sie OMVS ein

Verwenden Sie den folgenden Befehl, um OMVS einzugeben:

```
TSO OMVS
```

## Schritt 6: 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
```

> **Note**
>
> Der Einhängepunkt wird in Schritt 3 definiert.

## Schritt 7: 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 8: Erstellen Sie das Arbeitsverzeichnis

Verwenden Sie den folgenden Befehl, um die Arbeitsverzeichnisumgebung festzulegen:

```
mkdir -p $WORK_DIR
```

## Schritt 9: Kopieren Sie die Tar-Datei des Agenten und das Arbeitsverzeichnis

Laden Sie die Agent-Tar-Datei über den M2-Agent-Link von AWS herunter.

Der Übertragungsmechanismus hängt von Ihrer Umgebung ab. Stellen Sie jedoch sicher, dass die Tar-Datei im Binärmodus übertragen wird.

## Schritt 10: Beenden Sie die Agenteninstallation

Gehen Sie wie folgt vor, um die Agenteninstallation abzuschließen.

1. Stellen Sie die Umgebungsvariable m2-agent-Version mit dem folgenden Befehl auf die Version ein, 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-$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 abschickenCPY#PDS, um die Datensätze des File Transfer Agents zu erstellen.

   > **Note**
   >
   > JCL verwendet dieSYS2.AWS.M2 HLQ.

Um den File Transfer Agent zu erstellen, aktualisieren Sie die drei symbolischen Variablen HLQ (High Level Qualifier) und`V0LSER`, die später in der `AGNTPATH` JCL verwendet werden sollen:

```
oedit $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.

## Einen File Transfer Agent konfigurieren

Nachdem Sie einen File Transfer Agent installiert haben, gehen Sie wie folgt vor, um den Agenten zu konfigurieren. Wenn Sie einen neuen Agenten installieren müssen, folgen Sie den Anweisungen auf der the section called “Installieren Sie einen File Transfer Agent” Seite.

Themen

- Schritt 1: Berechtigungen konfigurieren und Task Control (STC) starten
- Schritt 2: Amazon S3 S3-Buckets erstellen
- Schritt 3: Erstellen Sie einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung
- Schritt 4: Erstellen Sie ein AWS Secrets Manager Geheimnis für die Mainframe-Anmeldeinformationen
- Schritt 5: Erstellen Sie eine IAM-Richtlinie
- Schritt 6: Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten
- Schritt 7: Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll
- Schritt 8: Agentenkonfiguration

# Schritt 1: Berechtigungen konfigurieren und Task Control (STC) starten

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.

   > **Note**
   >
   > `SYS2.AWS.M2`sollte durch den bei der Installation ausgewählten High-Level Qualifier (HLQ) ersetzt werden.

2. Aktualisieren Sie den PWD-Export in der `SYS2.AWS.M2.SAMPLIB(M2AGENT)` STC JCL, falls der Standard-Verzeichnispfad () für den File Transfer Agent geändert wurde. `/usr/lpp/aws/m2-agent`
3. Aktualisieren Sie den PROC gemäß den Standards Ihrer Website:
   a. Aktualisieren Sie die PROC-Karte gemäß Ihren Installationsanforderungen.
   b. Aktualisieren Sie die STEPLIB mit dem. `M2 LOADLIB PDSE ALIAS`
   c. Bearbeiten Sie PWD so, dass es auf den Installationspfad des Agenten verweist (nur dieser ist enthalten).
   d. `JAVA_HOME`Falls erforderlich, aktualisieren.
4. Aktualisieren und kopieren Sie die `SYS2.AWS.M2.SAMPLIB(M2AGENT)` JCL in `SYS1.PROCLIB` oder eine der Dateien PROCLIBs in Ihrer `PROCLIB` Verkettung.
5. Fügen Sie `SYS2.AWS.M2.LOADLIB` sie der APF-Liste mit dem folgenden Befehl hinzu:

   ```
   SETPROG APF ADD DSNAME(SYS2.AWS.M2.LOADLIB) SMS
   ```

6. Stellen Sie die Gruppe und den Besitzer des Agenten auf den Agenten (user/group (M2USER/M2GROUP) ein. Verwenden Sie im OMVS den folgenden Befehl:

   ```
   chown -R M2USER:M2GROUP $AGENT_DIR/current-version
   ```

> **Note**
>
> Bearbeiten Sie M2USER und M2GROUP mit den Namen, die Sie im Job mit den Sicherheitsdefinitionen verwendet haben.

## Schritt 2: Amazon S3 S3-Buckets erstellen

Für AWS Mainframe Modernization File Transfer ist ein Amazon S3 S3-Zwischenbucket als Arbeitsbereich erforderlich. Wir empfehlen, speziell dafür einen Bucket zu erstellen.

Erstellen Sie optional einen neuen Amazon S3 S3-Ziel-Bucket für die übertragenen Datensätze. Andernfalls können Sie auch Ihren vorhandenen Amazon S3 S3-Bucket verwenden. Weitere Informationen zum Erstellen von Amazon S3 S3-Buckets finden Sie unter Bucket erstellen.

## Schritt 3: Erstellen Sie einen vom AWS KMS Kunden verwalteten Schlüssel für die Verschlüsselung

Um einen vom Kunden verwalteten Schlüssel zu erstellen AWS KMS

1. Öffnen Sie die AWS KMS Konsole unterhttps://console.aws.amazon.com/kms.
2. Wählen Sie im linken Navigationsbereich vom Kunden verwaltete Schlüssel aus.
3. Klicken Sie auf Create key.
4. Wählen Sie unter Schlüssel konfigurieren die Option Schlüsseltyp als Symmetrisch und Schlüsselverwendung als Verschlüsseln und entschlüsseln aus. Verwenden Sie andere Standardkonfigurationen.
5. Wählen Sie Weiter aus.
6. Fügen Sie unter Labels hinzufügen einen Alias und eine Beschreibung für Ihren Schlüssel hinzu.
7. Wählen Sie Weiter aus.
8. Wählen Sie unter Definieren von Schlüsseladministratorberechtigungen mindestens einen IAM-Benutzer und eine IAM-Rolle aus, die diesen Schlüssel verwalten.
9. Wählen Sie Weiter aus.
10. Wählen Sie optional unter Wichtige Administratorberechtigungen definieren mindestens einen IAM-Benutzer und eine IAM-Rolle aus, die diesen Schlüssel verwenden können.
11. Wählen Sie Weiter aus.

12. Wählen Sie im Abschnitt Schlüsselrichtlinie bearbeiten die Option Bearbeiten aus und fügen Sie der Schlüsselrichtlinie die folgende Syntax hinzu. Dadurch kann der AWS Mainframe Modernization Service diese Schlüssel lesen und für die Verschlüsselung/Entschlüsselung verwenden.

> **Important**
>
> Fügen Sie die Anweisung zu den vorhandenen Anweisungen hinzu. Ersetzen Sie nicht, was bereits in der Richtlinie enthalten ist.

```
{
    "Sid" : "Enable AWS M2 File Transfer Permissions",
    "Effect" : "Allow",
    "Principal" : {
        "Service" : "m2.amazonaws.com"
    },
    "Action" : [
        "kms:Encrypt",
        "kms:Decrypt"
    ],
   "Resource" : "*"
},
```

13. Wählen Sie Weiter aus.
14. Überprüfen Sie auf der Seite „Überprüfen“ alle Details und wählen Sie „Fertig stellen“.

Kopieren und speichern Sie den ARN für den vom Kunden verwalteten Schlüssel, indem Sie den neu erstellten KMS-Schlüssel öffnen. Er wird später in der Richtlinie verwendet.

## Schritt 4: Erstellen Sie ein AWS Secrets Manager Geheimnis für die Mainframe-Anmeldeinformationen

Für den Zugriff auf die zu übertragenden Datensätze sind Mainframe-Anmeldeinformationen erforderlich, und diese müssen geheim gespeichert werden. AWS Secrets Manager

Um ein Geheimnis zu erstellen AWS Secrets Manager

1. Öffnen Sie die Secrets Manager-Konsole unterhttps://console.aws.amazon.com/secretsmanager.

2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Wählen Sie unter Geheimtyp auswählen die Option Anderer Geheimtyp aus.
4. Verwenden Sie den Schlüsselwert `userId` für die Mainframe-Benutzer-ID, die Zugriff auf die Datensätze hat. Verwenden Sie den Schlüsselwert `password` für das Passwortfeld.
5. Wählen Sie als Verschlüsselungsschlüssel den zuvor erstellten vom AWS Kunden verwalteten Schlüssel aus.
6. Wählen Sie Weiter aus.
7. Geben Sie auf der Seite „Geheimen Schlüssel konfigurieren“ einen Namen und eine Beschreibung ein.
8. Bearbeiten Sie auf derselben Seite die Ressourcenberechtigungen und verwenden Sie die folgende Ressourcenrichtlinie, damit der AWS Mainframe-Modernisierungsdienst darauf zugreifen kann.

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

9. Wählen Sie Speichern, um die aktualisierten Berechtigungen zu speichern.
10. Wählen Sie Weiter aus.
11. Gehen Sie zur Seite „Rotationen konfigurieren“ und wählen Sie „Weiter“.
12. Überprüfen Sie auf der Seite „Überprüfen“ alle Konfigurationen und wählen Sie „Speichern“, um das Geheimnis zu speichern.

> **Important**
>
> Bei den `userId` und `password` geheimen Schlüsseln wird zwischen Groß- und Kleinschreibung unterschieden und sie müssen wie abgebildet eingegeben werden.

# Schritt 5: Erstellen Sie eine IAM-Richtlinie

Um eine neue Richtlinie mit den für den Agenten erforderlichen Berechtigungen zu erstellen

1. Öffnen Sie unter https://console.aws.amazon.com/iam die IAM-Konsole.
2. Wählen Sie unter Zugriffsverwaltung die Option Richtlinien aus.
3. Wählen Sie Richtlinie erstellen aus.
4. Wechseln Sie auf der Seite Berechtigungen angeben unter Richtlinien-Editor vom visuellen Editor zum JSON-Editor und ersetzen Sie den Inhalt durch die folgende Vorlage:
5. 

```
{
"Version": "2012-10-17",
    "Statement": [
        {
         "Sid": "FileTransferAgentSQSReceive",
            "Effect": "Allow",
            "Action": [
                "sqs:DeleteMessage",
                "sqs:ReceiveMessage"
            ],
            "Resource": "arn:aws:sqs:*:111122223333:m2-*-request-queue.fifo"
        },
        {
         "Sid": "FileTransferAgentSQSSend",
            "Effect": "Allow",
            "Action": "sqs:SendMessage",
            "Resource": "arn:aws:sqs:*:111122223333:m2-*-response-queue.fifo"
        },
        {
         "Sid": "FileTransferWorkingS3",
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "<file-transfer-endpoint-intermediate-bucket-arn>/*"
        },
        {
         "Sid": "FileTransferAgentKMSDecrypt",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "<kms-key-arn>"
        }
    ]
```

```
}
```

6. Ersetzen Sie die ARNs `111122223333` in der Anforderungswarteschlange und der Antwortwarteschlange durch Ihr Konto.

> **Note**
>
> Dies sind Platzhalter-ARNs, die den beiden Amazon SQS SQS-Warteschlangen entsprechen, die während der Initialisierung des Datenübertragungsendpunkts erstellt wurden. Nachdem Sie einen Dateiübertragungsendpunkt erstellt haben, können Sie diese ARNs optional durch die tatsächlichen Werte von Amazon SQS ersetzen.

7. `file-transfer-endpoint-intermediate-bucket-arn`Ersetzen Sie es durch den ARN des zuvor erstellten Transfer-Buckets. Lassen Sie den Platzhalter „/*“ am Ende stehen.
8. `kms-key-arn`Ersetzen Sie durch den ARN des zuvor erstellten AWS KMS Schlüssels.
9. Wählen Sie Weiter aus.
10. Fügen Sie auf der Seite Überprüfen und erstellen den Namen und die Beschreibung der Richtlinie hinzu.
11. Wählen Sie Richtlinie erstellen aus.

## Schritt 6: Erstellen Sie einen IAM-Benutzer mit langfristigen Zugangsdaten

Erstellen Sie einen IAM-Benutzer, der es dem Mainframe-Agenten ermöglicht, eine Verbindung zu Ihrem Konto herzustellen. AWS Der Agent stellt eine Verbindung zu diesem Benutzer her und nimmt dann eine von Ihnen definierte Rolle mit den Berechtigungen an, die Antwort- und Anforderungswarteschlangen von Amazon SQS zu verwenden und Datensätze in Amazon S3 S3-Buckets zu speichern.

Um diesen IAM-Benutzer zu erstellen

1. Navigieren Sie zur IAM-Konsole unter. https://console.aws.amazon.com/iam
2. Wählen Sie unter Zugriffsverwaltung die Option Benutzer aus.
3. Wählen Sie Create user (Benutzer erstellen) aus.
4. Fügen Sie unter Benutzerdetails einen aussagekräftigen Benutzernamen hinzu. Beispiel, `Configure-ft-agent`.
5. Wählen Sie Weiter aus.

6. Wählen Sie in den Optionen für Berechtigungen die Option Richtlinien direkt anhängen aus, fügen Sie jedoch keine Berechtigungsrichtlinien an. Diese Berechtigungen werden von einer Rolle verwaltet, die angehängt wird.
7. Wählen Sie Weiter aus.
8. Überprüfen Sie die Details und wählen Sie Benutzer erstellen aus.
9. Sobald der Benutzer erstellt wurde, wählen Sie den Benutzer aus und öffnen Sie die Registerkarte Sicherheitsanmeldeinformationen.
10. Wählen Sie unter Zugriffsschlüssel die Option Zugriffsschlüssel erstellen aus.
11. Wählen Sie dann Andere, wenn Sie nach einem Anwendungsfall gefragt werden.
12. Wählen Sie Weiter aus.
13. Optional können Sie ein Beschreibungs-Tag wie, festlegen`Access key for configuring file transfer agent`.
14. Wählen Sie Zugriffsschlüssel erstellen aus.
15. Kopieren Sie den generierten Zugriffsschlüssel und den geheimen Zugriffsschlüssel und speichern Sie ihn sicher. Diese werden später verwendet.

Weitere Informationen zum Erstellen eines IAM-Zugriffsschlüssels finden Sie unter Zugriffsschlüssel für IAM-Benutzer verwalten.

> **Important**
>
> 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 und können später nicht abgerufen werden.

> **Note**
>
> Speichern Sie den IAM-Benutzer-ARN, der zum Einrichten einer Vertrauensbeziehung mit einer IAM-Rolle verwendet wurde.

# Schritt 7: Erstellen Sie eine IAM-Rolle, die der Agent übernehmen soll

Um eine neue IAM-Rolle für den Agenten zu erstellen

1. Wählen Sie Rollen in der IAM-Konsole unter. https://console.aws.amazon.com/iam
2. Wählen Sie Rolle erstellen aus.
3. Wählen Sie auf der Seite Vertrauenswürdige Entität auswählen die Option Benutzerdefinierte Vertrauensrichtlinie für den Entitätstyp Vertrauenswürdige aus.
4. Ersetzen Sie die benutzerdefinierte Vertrauensrichtlinie durch die folgende und `<iam-user-arn>` ersetzen Sie sie durch den zuvor erstellten ARN des Benutzers.

```
{
    "Version": "2012-10-17",
    "Statement": [ {
         "Sid": "FileTransferAgent",
         "Effect": "Allow",
         "Principal": {
            "AWS": "<IAM-User-arn>"
         },
         "Action": "sts:AssumeRole"
    } ]
}
```

5. Wählen Sie Weiter aus.
6. Filtern Sie unter Berechtigungen hinzufügen nach dem Richtliniennamen, den Sie zuvor erstellt haben, und wählen Sie ihn aus.
7. Wählen Sie Weiter aus.
8. Geben Sie der Rolle einen Namen und wählen Sie Rolle erstellen aus.

> **Note**
>
> Speichern Sie den Rollennamen, den Sie später zur Konfiguration des Mainframe-Agenten verwenden werden.

# Schritt 8: Agentenkonfiguration

Um den File Transfer Agent zu konfigurieren

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
   ```

   Zum Beispiel:

   ```
   agent.environments[0].account-id=<AWS_ACCOUNT_ID>
   agent.environments[0].agent-role-name=<AWS_IAM_ROLE_NAME>
   agent.environments[0].access-key-id=<AWS_IAM_ROLE_ACCESS_KEY>
   agent.environments[0].secret-access-id=<AWS_IAM_ROLE_SECRET_KEY>
   agent.environments[0].bucket-name=<AWS_S3_BUCKET_NAME>
   agent.environments[0].environment-name=<AWS_REGION>
   agent.environments[0].region=<AWS_REGION>
   zos.complex-name=<File_Transfer_Endpoint_Name>
   ```

   Wobei gilt:

   - `AWS_ACCOUNT_ID`ist die ID des AWS Kontos.
   - `AWS_IAM_ROLE_NAME`ist der Name der IAM-Rolle, die in der the section called “Schritt 7: 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 “Schritt 6: 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 “Schritt 6: 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**
>
> Sie können den File Transfer Agent in mehrere Regionen und Konten übertragen lassen, AWS indem Sie mehrere Umgebungen definieren.

- (Fakultativ). `zos.complex-name`ist der komplexe Name, den Sie beim Erstellen eines Dateiübertragungsendpunkts erstellt haben.

> **Note**
>
> Dieses Feld ist nur erforderlich, wenn Sie den komplexen Namen (der standardmäßig Ihren Sysplex-Namen verwendet) anpassen möchten, der dem entspricht, den Sie bei der Erstellung Ihres Dateiübertragungsendpunkts definiert haben. Weitere Informationen finden Sie unter the section called “Datenübertragungsendpunkte erstellen”.

> **Important**
>
> 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 so konfigurieren, dass er Daten in mehrere Regionen und Konten überträgt, AWS indem Sie Umgebungseinträge definieren.

> **Note**
>
> Ersetzen Sie die Werte durch die Parameterwerte, die Sie zuvor erstellt und konfiguriert haben.

```
#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
```

## Datenübertragungsendpunkte für die Dateiübertragung erstellen

Datenübertragungsendpunkte ermöglichen die Konnektivität mit dem Quell-Mainframe und unterstützen hohe Verfügbarkeit, Skalierbarkeit und eine optimierte Verwaltung von Agenten. Einzelne Agenten werden auf dem Mainframe installiert LPARs 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. Für die AWS Mainframe-Modernisierung gerüstet

# Datenübertragungsendpunkte erstellen

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-Modernisierungskonsole 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.

# Übertragungsaufgaben in File Transfer erstellen

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 Für die AWS Mainframe-Modernisierung gerüstet und erstellt the section called “Datenübertragungsendpunkte erstellen” 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. Auf der Seite Datensätze hinzufügen können Sie im Abschnitt Konfiguration der Übertragungsaufgabe wählen, ob Sie Ihre Übertragungsaufgabe entweder im Binärmodus konfigurieren oder Ihre Datensätze konvertieren und übertragen möchten.

   - Mit der Option „Übertragung im Binärmodus“ können Sie Datensätze übertragen, indem Sie Codepage-Konvertierungen überspringen und deren RDW (Record Descriptor Word) -Bytes beibehalten.
   - Mit der Option Datensätze übertragen und konvertieren können Sie Datensätze übertragen, indem Sie die Quell- und Zielcodepages für Ihre Datensätze festlegen. Sie können die verfügbaren Codepages für die Dateiübertragung auf der the section called “Unterstützte Quell- und Zielcodeseiten” Seite sehen.
7. Geben Sie 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.

   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.

   > **Note**
   >
   > 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.

8. Diese Datensätze werden in den Bereich Mainframe-Datensätze geladen, in dem 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. Wenn keine Datensätze geladen wurden, müssen Sie Schritt 7 erneut aufrufen.

> **Note**
>
> Sie können Datensätze aus mehreren Suchanfragen auswählen und sie zu Ihrer Übertragungsaufgabe hinzufügen.

9. Im Abschnitt Hinzugefügte Datensätze werden Name, Typ und Datenträgername Ihrer Datensätze angezeigt.

> **Important**
>
> Für die Option Datensätze übertragen und konvertieren müssen Sie die Quellcodepage und die Zielcodepage für jeden von Ihnen ausgewählten Datensatz manuell eingeben. 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.

10. Nachdem Sie die Datensätze im Abschnitt Hinzugefügte Datensätze (und die Quell- und Zielcodepages für die Option Datensätze übertragen und konvertieren) bestätigt haben, wählen Sie Weiter.
11. Auf der Seite Überprüfen und erstellen können Sie die Informationen für Ihre Übertragungsaufgabe überprüfen oder bearbeiten.
12. Wählen Sie dann Übertragungsaufgabe erstellen aus.

> **Important**
>
> Wenn Sie auf die Schaltfläche Übertragungsaufgabe erstellen klicken, wird die Datenübertragung gestartet, die gemäß der Preisübersicht für AWS Mainframe-Modernisierung abgerechnet werden kann. Diese Abrechnung basiert auf der übertragenen Datenmenge (GB), gemessen an der Größe des Datensatzes.

## 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-Modernisierungskonsole 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 sich Details zu jeder Übertragungsaufgabe aus dieser Liste anzeigen lassen 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 “Konfigurieren Sie einen Dateiübertragungsagenten” 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 “Datenübertragungsendpunkte erstellen” Seite, um einen neuen Datenübertragungsendpunkt zu erstellen.

### Schritt 4: Erstellen Sie eine Übertragungsaufgabe

Folgen Sie den Schritten auf der the section called “Erstellen Sie Übertragungsaufgaben” 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.

## Unterstützte Quell- und Zielkodierungen in AWS Mainframe Modernization File Transfer

AWS Mainframe Modernization File Transfer unterstützt verschiedene Datensatztypen und Optionen zur Codepage-Konvertierung.

### 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","", ", „GBK“, „IBM00858", „GB2312IBM01140", „IBM01141", „IBM01142", „IBM01143", „IBM01143", „IBM01144", „IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145",

„IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145", „IBM01145" 6", „IBM01147", „IBM01148", „IBM01149", „IBM037"," IBM1 026", "IBM1047","," ","," "0","," "0","," "0","," "0","," "IBM273IBM277IBM278IBM28IBM284IBM285IBM29IBM297IBM42IBM424IBM437IBM5IBM775IBM85IBM85 IBM855"," IBM857 "," IBM86 0", IBM861 ","," IBM862 "," IBM863 "," IBM864 "," IBM865 ",IBM866", ""“, IBM868 ",IBM869", ",IBM87", 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_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“, „SHIFT_JIS“, „TIS_620", „US_ASCII“, „UTF_16", „UTF_16BE“, „UTF_16LE“, „UTF_32BE“, „UTF_32BE“, „UTF_32LE“, „UTF_8", „WINDOWS_1250", „WINDOWS_1251", „WINDOWS_1252", „WINDOWS _1253", "WINDOWS_1254", „WINDOWS_1255", „WINDOWS_1256", „WINDOWS_1257", „WINDOWS_1258", „WINDOWS_31J“, „BIG5X_HKSCS_2001", „BIG5X__SOLARIS“, „X_EUCJP_OPEN“, „X_EUC_JP_LINUX“, „X_EUC_JP_LINUX“, „X_EUC_TW“ _ IBM1 006", „X_ IBM1 025", „X_ IBM1 046", „X_ IBM1 097", „X_ IBM1 098", „X_", „X_“, „X_“, „X_IBM1112“, „X_IBM1122“ IBM1123 IBM1124", „X_IBM1129", „X_IBM1166", „X_", „X_IBM1364“, „X_IBM1381", „X_ IBM29626 C“, „X_ IBM1383 IBM3 00", „X_", „X_", „X_IBM33722", „X_IBM737", „X_", „X_IBM833“, „X_IBM834“, „X_IBM856“, „X_“, „X_IBM874“, „X_IBM875“, „X_IBM921“, „X_IBM922“, „X_“, „X_IBM93“, „X_IBM933“, „X_IBM935“, „X_“, „X_IBM937“, „X_IBM939“, „X_IBM942“, „X_“, „IBM942X_C“, „X_IBM943“, „IBM943X_C“, „X_IBM948“, „X_“, „X_IBM949“, „IBM949X_C“, „X_“ IBM95 0", „X_IBM964", „X_ IBM97 0", „X_ISCII91", „X_ISO_2022_CN_CNS“, „X_ISO_2022_CN_GB“, „X_ISO_8859_11", „X_JIS0208", „X_JISAUTODETECT“, „X_JOHAB“, „X_MACARABIC“, „X_MACCENTRAL“ EUROPA“, „X_MACCROATIAN“, „X_MACCYRILLIC“, „X_MACDINGBAT“, „X_MACGREEK“, „X_MACHEBREW“, „X_MACICELAND“, „X_MACROMAN“, „X_MACROMANIA“, „X_MACSYMBOL“, „X_MACTHAI“, „X_MACTURKISH“, „X_MACUKRAINE“, „MS932X_0213", „MS95X_0_HKSCS“, „MS95X_0_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", „ISO2X_WINDOWS_022j“

# AWS Transform zur Modernisierung von Mainframe-Anwendungen

AWS Transform for Mainframe ermöglicht es Ihnen, ältere COBOL-Mainframe-Anwendungen schneller auf Java-Anwendungen umzustellen, indem Sie die komplexen, zeitaufwändigen Aufgaben der Analyse von Codebasen, der Planung der Transformation, der Generierung der technischen Dokumentation, der Extraktion der Geschäftslogik, der Codezerlegung, der Wellenplanung und der Ausführung von Refactoring und Reforging automatisieren. Es reduziert die Kosten und die Komplexität der Mainframe-Modernisierung, indem es generative KI und Automatisierung nutzt und gleichzeitig Ihre geschäftskritische Geschäftslogik beibehält. AWS Transform Dank der natürlichen Sprachschnittstelle und des zielorientierten Ansatzes behalten Sie die Kontrolle über die Transformation. So können Sie sich auf strategische Prioritäten konzentrieren, während die Automatisierung den Großteil der Modernisierung übernimmt.

Weitere Informationen zu den Funktionen und wichtigsten Funktionen, einer umfassenden Komplettlösung und der Beteiligung von Mitarbeitern an Eingaben und Verarbeitung finden Sie im Benutzerhandbuch unter Migration und Modernisierung von Mainframe-Anwendungen.AWS Transform

## Wichtigste Vorteile

AWS Transform Die Funktionen zur Modernisierung von Mainframe-Anwendungen bieten mehrere Vorteile. Einige davon beinhalten:

- Beschleunigen Sie die Mainframe-Modernisierung mit generativer KI: AWS Transform hilft Ihnen, Ihren COBOL-Code innerhalb weniger Monate in modernen Java-Code umzuwandeln, anstatt der traditionelle Zeitplan in Jahren.
- Wissenslücken schließen: AWS Transform Generiert eine umfassende Dokumentation für Ihre Mainframe-Anwendungen, schließt die Wissenslücke und ermöglicht fundiertere Entscheidungen.
- Behalten Sie die kritische Geschäftslogik bei: AWS Transform Behält die kritische Geschäftslogik Ihrer Altsysteme bei und stellt sie gleichzeitig auf moderne, Cloud-optimierte Java-Anwendungen um.
- Geschäfts- und technische logische Domänenzerlegung: Zerlegt die Mainframe-Codebasis AWS Transform automatisch in verschiedene Geschäftsbereiche, um den manuellen Aufwand und den Zeitaufwand für die Codebase-Analyse und -Zerlegung zu reduzieren.

- HITL-Funktionen (Human in the Loop): AWS Transform bietet einen autonomen, zielorientierten Ansatz, mit dem Sie die Kontrolle über die Mainframe-Modernisierung behalten.

# Komplettlösung zur Transformation der Mainframe-Anwendungskonsole

In der AWS Transform Weberfahrung können Sie die Transformation Ihrer Mainframe-Anwendungen von COBOL nach Java durchführen. Um zu verstehen, wie diese Funktion verwendet wird, folgen Sie allen Schritten auf der Seite Transformation von Mainframe-Anwendungen im AWS Transform Benutzerhandbuch.

# 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 EC2 Amazon-Instances und schaffen eine maßgeschneiderte Umgebung für die Datenreplikation von Mainframe-Systemen bis hin zur AWS Verwendung von Precisely. Dieses Handbuch enthält die erforderlichen Schritte, um auf diese zuzugreifen und sie zu verwenden. AMIs

## Voraussetzungen

- Stellen Sie sicher, dass Sie Administratorzugriff auf ein AWS Konto haben, mit dem Sie EC2 Amazon-Instances erstellen können.
- Stellen Sie sicher, dass der AWS Mainframe-Modernisierungsservice in der Region verfügbar ist, in der Sie die EC2 Amazon-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 EC2 Amazon-Instances erstellt werden.
- Stellen Sie beim Erstellen von EC2 Amazon-Instances in einer Amazon VPC sicher, dass die zugehörige Routing-Tabelle über ein Internet-Gateway oder ein NAT-Gateway verfügt.

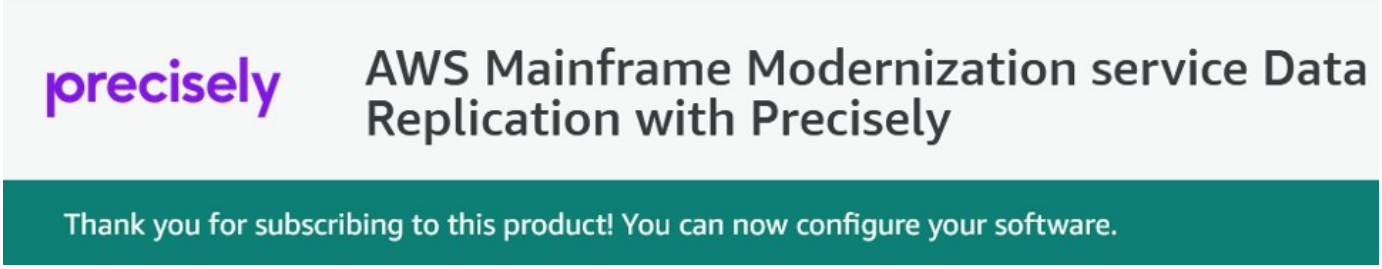
> **Note**
>
> Eine erfolgreiche Datenreplikation setzt voraus, dass die EC2 AWS-Instance Kommunikationszugriff auf den AWS Marketplace hat. Wenn es ein Verbindungsproblem mit dem AWS Marketplace gibt, schlägt der Replikationsprozess fehl.

## 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 https://console.aws.amazon.com/marketplace an AWS Management Console und öffnen Sie die AWS Marketplace Konsole.

2. Wählen Sie Manage subscriptions (Abonnements verwalten).
3. Navigieren Sie je nach Anwendungsfall zu einem der folgenden Links:
   - Datenreplikation für IBM z/OS: https://aws.amazon.com/marketplace/pp/prodview-doe2lroefogia
   - Datenreplikation für IBM i: https://aws.amazon.com/marketplace/pp/prodview-iqrkflccxf7ko
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 die Dankesnachricht angezeigt wird, wie unten gezeigt. 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 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, für die die Zulassung gilt.
5. Wählen Sie Weiter, um zu starten. EC2 Mit dieser Aktion gelangen Sie zur EC2 Amazon-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 EC2 Amazon-Schlüsselpaare und Linux-Instances im EC2 Amazon-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 Precisely EC2 Serverinstanz ist es in der Regel nicht nur möglich, SSH-Zugriff (standardmäßig auf Port 22) zu ermöglichen, sondern auch TCP-Verkehr über den Standardport 2626 zuzulassen.
11. Konfigurieren Sie den Speicher für die EC2 Amazon-Instance.
12. Lesen Sie die Zusammenfassung 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 EC2 Amazon-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 EC2 Mainframe-Modernisierungs-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 EC2 Amazon-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 präzise Abrechnung unerlässlich ist. Wenn diese Konfiguration nicht implementiert wird, kann dies zu erfolglosen Datenreplikationsversuchen und Betriebsunterbrechungen führen.

1. Öffnen Sie unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
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 unter https://console.aws.amazon.com/iam/ die IAM-Konsole.
2. Wählen Sie im Navigationsbereich Rollen und dann 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 aus EC2.
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 EC2 Amazon-Instance an

1. Öffnen Sie die EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
2. Wählen Sie im Navigationsbereich Instances aus.
3. Wählen Sie Ihre EC2 Amazon-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.

Weitere Informationen zu den ersten Schritten mit der AWS Datenreplikation für IBM i finden Sie unter Überblick über die Datenreplikation.

# AWS Mainframe Modernization Code-Konvertierung mit mLogica

AWS Mainframe Modernization Die Codekonvertierung mit mLogica (Codekonvertierung) ist eine AWS Mainframe Modernization Funktion, die z/OS Mainframe-Assembler-Code automatisch in COBOL konvertiert. Sie können die Codekonvertierung verwenden, um ein Assembler-Image abzurufen, indem Sie den AWS CodeBuild Dienst für Ihre beabsichtigte Codekonvertierung mit Ihrem verwenden. AWS-Konto

Themen


- Was ist Assembler-Konvertierung mit mLogica?
- Verstehen Sie die Abrechnung der Codekonvertierung für die Assembler-Konvertierung
- Konzepte zur Codekonvertierung
- Verstehen Sie die Komponenten und Prozesse für die Codekonvertierung
- Tutorial: Code von Assembler nach COBOL konvertieren in AWS Mainframe Modernization


## Was ist Assembler-Konvertierung mit mLogica?

AWS Mainframe Modernization Die Codekonvertierung mit mLogica (Codekonvertierung) konvertiert automatisch z/OS Mainframe-Assembler-Code in COBOL. Der Dienst läuft innerhalb Ihres AWS-Konto und überträgt oder speichert keinen Assembler- oder COBOL-Quellcode außerhalb von. AWS-Konto Die Codekonvertierung ermöglicht es Ihrem autorisierten Konto, mithilfe des AWS CodeBuild Dienstes ein Assembler-Image für Ihre beabsichtigte Codekonvertierung abzurufen.

AWS Mainframe Modernization bietet Ihnen die Möglichkeit, Builds und Continuous integration/continuous Delivery (CI/CD) -Pipelines für Ihre migrierten Anwendungen einzurichten. Diese Builds und Pipelines verwenden AWS CodeBuild Amazon S3, um diese Funktion bereitzustellen. AWS CodeBuild ist ein vollständig verwalteter Build-Service, der Ihren Quellcode kompiliert, Komponententests durchführt und Artefakte erzeugt, die sofort einsatzbereit sind. Amazon S3 ist ein Objektspeicherservice, der branchenführende Skalierbarkeit, Datenverfügbarkeit, Sicherheit und Leistung bietet.

Themen


- Compiler für die Codekonvertierung

- Architektur für die Codekonvertierung
- Automatisierungsansatz
- Sicherheit
- Weitere Ressourcen

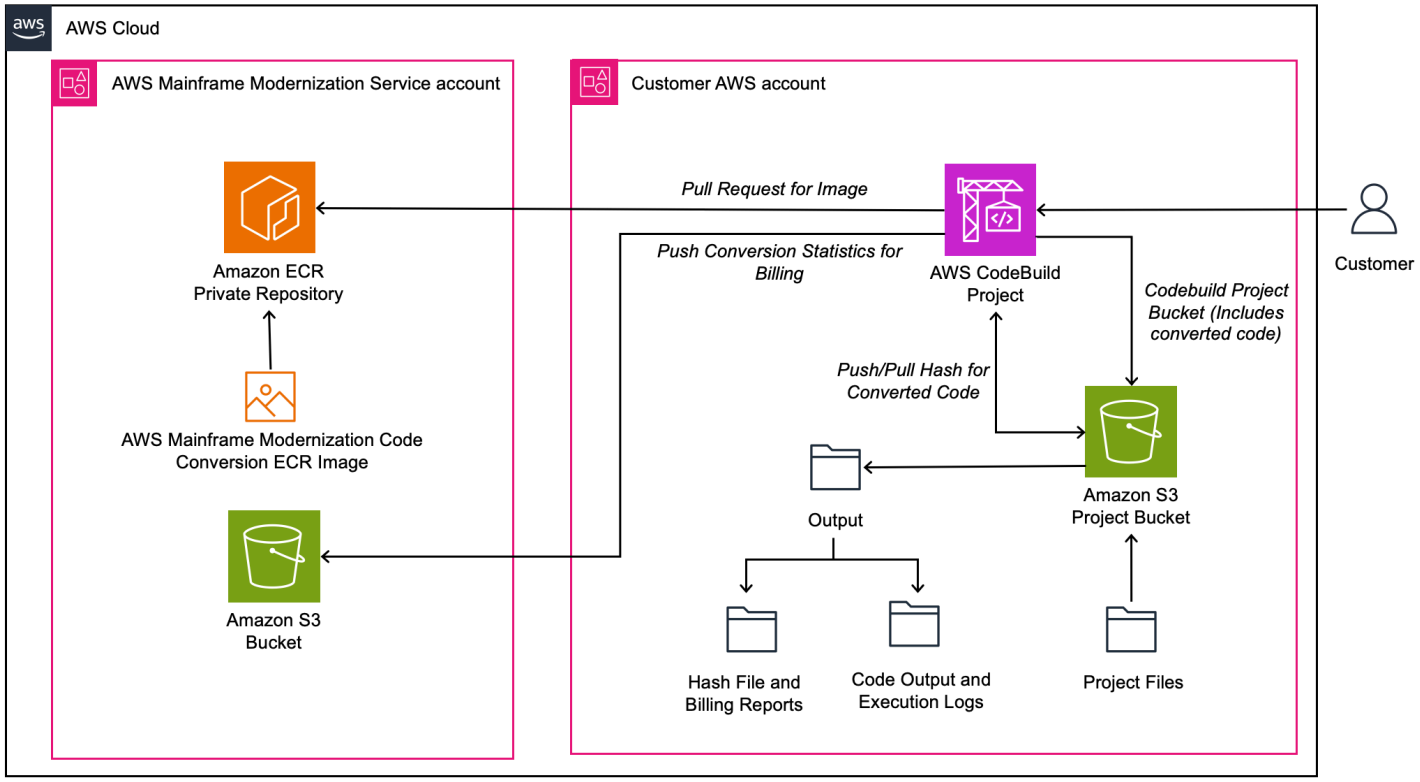

## Compiler für die Codekonvertierung

Die Codekonvertierung kann so konfiguriert werden, dass COBOL ausgegeben wird, das für die Kompilierung und Ausführung in mehreren Zielumgebungen mit unterschiedlichen Compilern geeignet ist. Einige davon beinhalten:

- M2 Re-Platforming mit Rocket Software (ehemals Micro Focus) und anderen Rocket Enterprise Server-Umgebungen
- M2-Neuausrichtung mit NTT DATA Enterprise COBOL () UniKix
- mLogica LIBER*COBOL
- z/OSMainframe mit IBM Enterprise COBOL
- Sehr wichtig ist COBOL

## Architektur für die Codekonvertierung

Das Folgende ist ein Architekturdiagramm für den Codekonvertierungsprozess:




## Automatisierungsansatz

Um die Codekonvertierung mit verwenden zu können CodeBuild, muss der Assembler-Code in einen Amazon S3 S3-Bucket hochgeladen werden, um später die Konvertierungsparameter zu konfigurieren und ein CodeBuild Projekt aufzurufen, um jeden Schritt des Konvertierungsprozesses durchzuführen. Der COBOL-Zielcode wird automatisch in einem angegebenen Pfad im Amazon S3 S3-Bucket gespeichert.

## Sicherheit

AWS Mainframe Modernization Die Codekonvertierung ermöglicht die Konvertierung unter Beibehaltung des gesamten Quell- und Zielcodes in Ihrem AWS-Konto. Der Quell-Assembler-Code, der Ziel-COBOL-Code und die Konfigurationsdateien werden in Ihrem Amazon S3 S3-Bucket gespeichert. Das automatisierte Konvertierungstool wird als Container in Ihrer CodeBuild Umgebung ausgeführt. AWS-Konto Der Code bleibt jederzeit in Ihrem Konto.

Damit das Konvertierungstool auf Ihren Amazon S3 S3-Bucket zugreifen kann, erteilen Sie einer AWS-Service Rolle Berechtigungen für den Bucket. Bei der Konfiguration legen Sie diese Servicerolle so fest CodeBuild, dass sie auf das Container-Image und auf Ihren Amazon S3 S3-Bucket zugreifen CodeBuild kann.

## Weitere Ressourcen

Darüber hinaus finden Sie hier einige zusätzliche Ressourcenthe section called “Tutorial: Code von Assembler nach COBOL konvertieren”, in denen Sie mehr über die Erstellung der AWS CloudFormation Vorlagen und weitere Informationen zur Konvertierung von Assembler in COBOL erfahren können.

- Workshop-Link für die automatisierte Codekonvertierung von Assembler nach COBOL:. https://catalog.workshops.aws/awsm2ccm-assembler-cobol/en-US
- Blogbeitrag:. https://aws.amazon.com/blogs/migration-and-modernization/unlocking-new-potential-transform-your-assembler-programs-to-cobol-with-aws-mainframe-modernization/

# Verstehen Sie die Abrechnung der Codekonvertierung für die Assembler-Konvertierung

Auf dieser Seite erfahren Sie mehr über den Umfang und das Verfahren zur Abrechnung der Code-Konvertierung, bevor Sie die eigentliche Konvertierung durchführen. Im Abschnitt zur Rechnungsberechnung wird der Prozess erwähnt, bei dem die Konvertierung von Assembler zu COBOL pro Codezeile berechnet wird.

## Abrechnung und Umfang der Codekonvertierung

Bei der Assembler-Code-Konvertierung werden Ihnen AWS-Konto erst nach Abschluss des Konvertierungsschritts Gebühren (Abrechnungsberichte) berechnet. Die Gebühr basiert auf der Anzahl der konvertierten Codezeilen. Wenn Sie mehrere Konvertierungsschritte durchführen, z. B. nachdem Sie neuen Assembler-Code hinzugefügt, die Konvertierungskonfiguration geändert oder eine neue Version des Containers angewendet haben, werden nur geänderte Zeilen und/oder neu hinzugefügte Zeilen zur Berechnung der Gebühr verwendet. Wir berechnen Ihnen nicht zweimal die Konvertierung derselben Codezeile im selben Programm.

> **Note**
>
> Die Module mit geänderten Codezeilen und alle Codezeilen in neuen oder umbenannten Programmen werden berechnet.

Um mehrere Gebühren zu vermeiden, speichert die Codekonvertierung eine kodierte Binärdatei für jedes Assembler- oder Macro-Modul im Projekt-Bucket in. `<Project_bucket>/awsm2ccm-do-not-delete/<AWS_account_number>/Hash` Diese codierten Dateien enthalten keinen Kundencode.

> **Important**
>
> Bearbeiten oder löschen Sie diese Dateien nicht manuell. Änderungen können zu mehreren Fakturierungen für die Konvertierung derselben Komponenten führen.

Der Analysebericht zur AWS Mainframe Modernization Code-Konvertierung („Analysebericht“) bietet Kunden Informationen über den voraussichtlichen Umfang, das Ergebnis und die Abrechnung der Konvertierung, um sicherzustellen, dass die tatsächlichen Konversionserwartungen korrekt sind. Die Konvertierung kann dazu führen, dass einige Codezeilen nicht, einige Codezeilen teilweise und einige Codezeilen vollständig konvertiert werden. Der Analysebericht zeigt die Anzahl der Codezeilen für jede Kategorie. Vor der Konvertierung von Programmen, Makros und Copybooks müssen Sie den Analysebericht ausführen und lesen. Sobald ein Kunde den Analysebericht überprüft und mit dem gemeldeten Umfang, dem erwarteten Ergebnis und der erwarteten Abrechnung einverstanden ist, kann er mit der Durchführung der Konvertierung fortfahren.

> **Note**
>
> Durch die Ausführung des **`Convert`** Befehls AWS Mainframe Modernization Code Conversion bestätigen Sie, dass Sie den Analysebericht ausgeführt und gelesen haben und dass Sie mit dem erwarteten Ergebnis und der fakturierbaren Anzahl von Codezeilen einverstanden sind.

## Umfang der Konvertierung

AWS Mainframe Modernization Bei der Codekonvertierung werden alle Codezeilen aller Assembler-, Makro- und Copybook-Komponenten verarbeitet, die in den Verzeichnissen scrlib und macrolib im konfigurierten S3-Quellverzeichnis verfügbar sind. Assembler-Programme sowie alle Makros und Copybooks, auf die in einem Assembler-Programm verwiesen wird, fallen in den Geltungsbereich. Makro- und Copybook-Komponenten, auf die kein Assembler-Programm verweist, gelten als außerhalb des Gültigkeitsbereichs liegend und werden nicht konvertiert. Während der Verarbeitung führt der Konverter erweiterte Algorithmen aus, die jede im Gültigkeitsbereich enthaltene Komponente

ganzheitlich betrachten. Alle Codezeilen dieser Komponenten sind an der Verarbeitung beteiligt, unabhängig davon, ob sie vollständig, teilweise oder nicht konvertiert wurden. AWS Mainframe Modernization Bei der Codekonvertierung werden Leerzeilen ignoriert und nicht als Codezeilen gezählt. Kommentarzeilen und Zeilen, die anderen Text enthalten (z. B. in JCL eingebettete JCL-Anweisungen für Assembler), werden bei der Abrechnung als Codezeilen gezählt.

## Berechnung der Abrechnung

AWS Mainframe Modernization Gebühren für die Codekonvertierung für alle im Leistungsumfang enthaltenen Komponenten. Das bedeutet, dass für jede Codezeile innerhalb jeder im Geltungsbereich enthaltenen Komponente Gebühren berechnet werden, einschließlich Zeilen, die nicht konvertiert werden konnten, teilweise konvertiert wurden und vollständig konvertiert wurden. AWS Mainframe Modernization Bei der Codekonvertierung werden alle Codezeilen der zur Verarbeitung bereitgestellten Komponenten (einschließlich Assembler-Programme, referenzierter Copybooks und referenzierter Makros) summiert und die Gesamtzahl der Codezeilen für die Abrechnung verwendet.

> **Note**
>
> Copybooks und Makros, auf die in einem Assembler-Programm nicht verwiesen wird, fallen nicht in den Geltungsbereich.

Nehmen wir zum Beispiel an, ein Programm hat 1.000 Codezeilen:

- 700 Zeilen sind vollständig konvertiert
- 200 Zeilen werden teilweise konvertiert
- 100 Zeilen werden nicht konvertiert

1.000 Codezeilen würden verarbeitet und wären abrechnungsfähig.

## Verbesserung der Konvertierung

Wenn Sie als Kunde eine höhere Konversionsrate für die Codezeilen anstreben oder andere spezifische Anforderungen haben, können Sie sich an die AWS Mitarbeiter wenden, um zusätzliche Optionen wie Kalibrierungsmaßnahmen oder professionelle Unterstützung zu erhalten.

# Konzepte zur Codekonvertierung

Um zu erfahren, wie die Codekonvertierung abläuft, ist es wichtig, einige wichtige Konzepte zu verstehen, z. B. die Handhabung von Makros, Codepages und CodeBuild .

Themen


- Umgang mit Makros
- Codepages (EBCDIC oder ASCII)
- CodeBuild


## Umgang mit Makros

Mainframe-Assembler-Code verwendet häufig Makros, um Funktionen für die Wiederverwendung zu kapseln. Das Verhalten von Makros wird in der Regel zur Laufzeit der Anwendung anhand von Parametern bestimmt, die von einem Assembler-Programm übergeben werden. Die Codekonvertierung bietet mehrere Mechanismen zur Erweiterung von Assembler-Makros vor der Konvertierung in COBOL.

## Codepages (EBCDIC oder ASCII)

Mainframe-Assembler enthalten häufig Zeichenliterale, ausgedrückt als Hexadezimalwerte, die EBCDIC-Zeichen entsprechen. Die Codekonvertierung bietet eine konfigurierbare Funktion zur automatischen Verwaltung von Zeichenliteralen in ASCII, wenn COBOL für ASCII-Umgebungen ausgegeben wird.

## CodeBuild

Die Codekonvertierung ist über den Service verfügbar. AWS CodeBuild AWS CodeBuild ist ein Tool zur Build-Automatisierung, das ursprünglich als Teil einer CI/CD Pipeline entwickelt wurde. In AWS CodeBuild wird verwendet AWS Mainframe Modernization, um das MCCAC-Konvertierungstool und andere Tools wie den COBOL-Compiler von Rocket Software (ehemals Micro Focus) zu automatisieren.

# Verstehen Sie die Komponenten und Prozesse für die Codekonvertierung

AWS Mainframe Modernization Der Prozess der Codekonvertierung umfasst verschiedene Komponenten wie AWS Mainframe Modernization Container, S3-Projekt-Bucket und Speicherorte für Protokolldateien.

Themen


- AWS Mainframe Modernization Container
- S3-Projekt-Bucket
- Speicherorte der Protokolldateien
- Prozessübersicht


## AWS Mainframe Modernization Container

AWS Mainframe Modernization Der Codekonvertierungscontainer wird im AWS CodeBuild Projekt ausgeführt und bietet Befehle zum Einrichten der Projektverzeichnisse und Konfigurationsdateien, zum Bewerten von Assembler-Code, zum Erweitern von Assembler-Makros und zum Konvertieren von Assembler-Code in COBOL.

Sie werden Zugriff auf das folgende AWS ECR Repository haben:`381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod`.

Um die Bilder zu verwenden, können Sie eine der beiden Optionen wählen:

- Verwenden Sie das neueste Tag, wenn Sie das Bild über abrufen AWS CodeBuild. Wenn Sie das Bild verwenden, verwenden Sie diesen Pfad:`381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod`. Das bedeutet, AWS CodeBuild dass das Bild, das zuletzt in das Repository übertragen wurde, übernommen wird.
- Die Version auflisten und aus dieser auswählen. Verwenden Sie dazu den folgenden Befehl über CLI, um die verschiedenen Versionen im Repository aufzulisten:

```
aws ecr describe-images \
  --registry-id 381492161314 \
  --repository-name aws-mlogica-codebuild-prod \
  --query 'imageDetails[*].{ImagePushedAt: imagePushedAt, ImageTags: imageTags}' \
```

```
 --output json | jq '[.[] | {ImageURI: (.ImageTags[] |
"381492161314.dkr.ecr.us-east-1.amazonaws.com/aws-mlogica-codebuild-prod:" + .),
ImagePushedAt: .ImagePushedAt}] | sort_by(.ImagePushedAt) | reverse'
```

Dadurch werden alle Bilder mit dem zugehörigen Tag auf jedem Bild sowie der Zeitpunkt, zu dem ein bestimmtes Bild im Repository veröffentlicht wurde, aufgelistet. Basierend auf dem obigen Code erhalten Sie eine Liste von Bildern, wobei das Tag auf dem Bild die Version des Tools zur Codekonvertierung darstellt. Sie können das entsprechende Bild entsprechend Ihren Anforderungen auswählen.

## S3-Projekt-Bucket

Der Eingabe- und Ausgabecode, der mit erweiterten Makros aktualisierte Code und die durch die AWS Mainframe Modernization Codekonvertierung generierten Berichte werden in dem Projekt-Bucket gespeichert, den Sie in Ihrem AWS -Kontenverwaltung erstellen. Sie gewähren der AWS Mainframe Modernization Codekonvertierung Zugriff auf den Bucket, indem Sie einer AWS Servicerolle Berechtigungen erteilen.

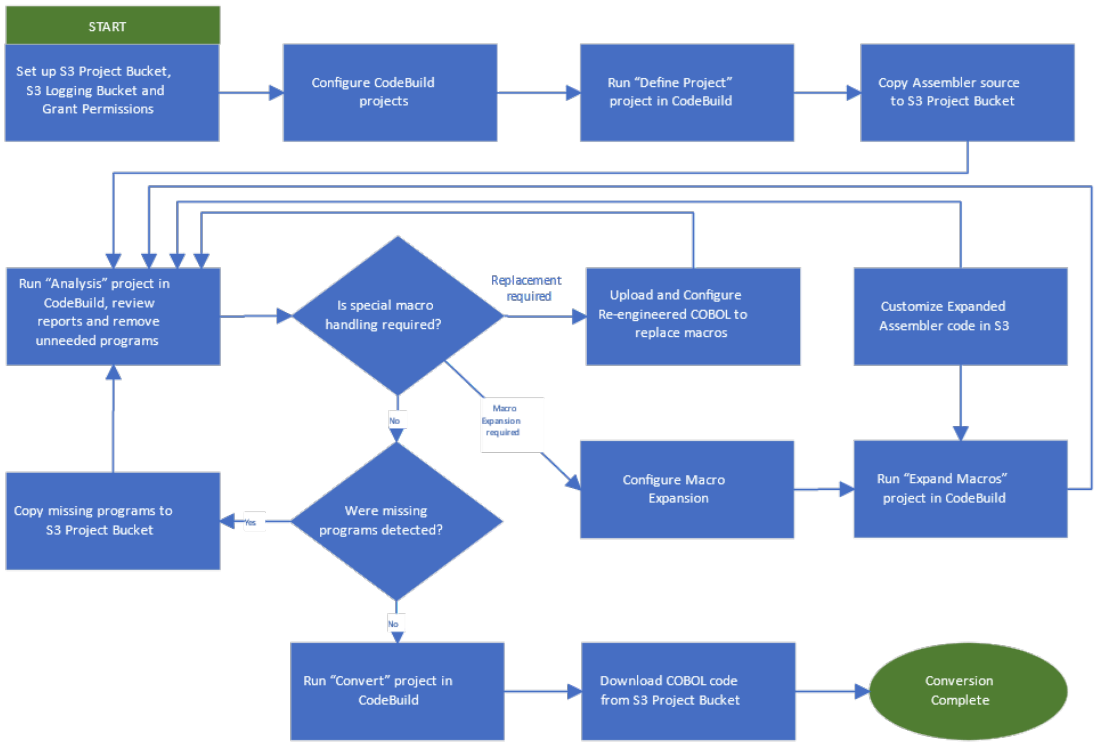
## Speicherorte der Protokolldateien

Protokolldateien werden bei jeder CodeBuild Projektausführung an zwei Speicherorten geschrieben:

- Protokolldateien mit allgemeinen Ergebnissen der einzelnen CodeBuild Schritte werden in Protokolldateien im Logging-Bucket geschrieben, der im konfiguriert ist CodeBuild. Diese Dateien erscheinen als GZIP-Archive mit einem vom CodeBuild Framework generierten Dateinamen vom Typ GUID (z. B.`0c03e183-ab40-4fe0-ba77- bc1d87e73b14.gz`). Jedes Archiv enthält das Protokoll, das bei der Ausführung eines CodeBuild Projekts generiert wurde. Wenn die Ausführung eines CodeBuild Projekts fehlschlägt, enthält diese Protokolldatei wichtige Informationen zur Fehlerbehebung.
- Protokolldateien mit detaillierten Ausführungsergebnissen auf Komponentenebene werden mit dem Dateinamenmuster `<Project_Bucket_name>_.log` (z. B.`project-bucket_202406131200.log`) in die Protokolldateien im Haupt-Bucket-Pfad des Projekts geschrieben. Diese Protokolle bieten:
  - Eine Zusammenfassung der Konfiguration mit Angabe der Eingabe- und Ausgabeorte.
  - Ein Protokoll aller verarbeiteten Assembler- oder Macro-Komponenten mit dem Zieldateinamen.
  - Eine Liste von Berichten, die mit Dateispeicherorten generiert wurden.
  - Bei Konvertierungsausführungen eine Liste der mitgelieferten Runtime-Copybooks.

## Prozessübersicht

Das folgende Diagramm veranschaulicht den Prozess der Konvertierung von Assembler nach COBOL:




# Tutorial: Code von Assembler nach COBOL konvertieren in AWS Mainframe Modernization

Sie können dieses Dokument als step-by-step Leitfaden verwenden, um zu verstehen, wie der Assembler-Code für die Mainframe-Modernisierung nach COBOL konvertiert wird. Darüber hinaus finden Sie im Workshop Automatisierte Codekonvertierung von Assembler zu COBOL weitere Informationen über den Konvertierungsprozess.

Themen


- Voraussetzungen
- Schritt 1: Teilen Sie die Build-Assets mit AWS-Konto
- Schritt 2: Amazon S3 S3-Buckets erstellen
- Schritt 3: IAM-Richtlinie erstellen
- Schritt 4: Erstellen Sie eine IAM-Rolle

- Schritt 5: Hängen Sie die IAM-Richtlinie an die IAM-Rolle an
- Schritt 6: Erstellen Sie das CodeBuild Projekt
  - Schritt 6.1: Erstellen Sie das Define-Projekt
  - Schritt 6.2: Erstellen Sie das Codeanalyse-Projekt
  - Schritt 6.3: Erstellen Sie das Code-Konvertierungsprojekt
- Schritt 7: Definieren Sie das Projekt und laden Sie den Quellcode hoch
- Schritt 8: Führen Sie die Analyse durch und verstehen Sie die Berichte
- Schritt 9: Führen Sie die Codekonvertierung aus
- Schritt 10: Überprüfen Sie die Codekonvertierung
- Schritt 11: Laden Sie den konvertierten Code herunter
- Bereinigen von -Ressourcen


## Voraussetzungen

Lesen Sie den Verstehen Sie die Abrechnung der Codekonvertierung für die Assembler-Konvertierung Abschnitt, um zu erfahren, wie die Assembler-Codekonvertierung Gebühren (Abrechnungsberichte) für Sie AWS -Kontenverwaltung generiert und wie die Abrechnung funktioniert.

## Schritt 1: Teilen Sie die Build-Assets mit AWS-Konto

Stellen Sie in diesem Schritt sicher, dass Sie die Build-Assets mit Ihnen teilen AWS-Konto, insbesondere in der Region, in der die Assets verwendet werden.

1. Öffnen Sie die AWS Mainframe Modernization Konsole unter https://console.aws.amazon.com/m2/.
2. Wählen Sie in der linken Navigationsleiste Tools aus.
3. Wählen Sie in AWS Mainframe Modernization Code Conversion with mLogica die Option Share assets with my. AWS-Konto

> **Important**
>
> Sie müssen diesen Schritt in jeder AWS Region, in der Sie Builds durchführen möchten, einmal ausführen.

## Schritt 2: Amazon S3 S3-Buckets erstellen

In diesem Schritt erstellen Sie Amazon S3 S3-Buckets. Der erste Bucket ist der Projekt-Bucket AWS CodeBuild , in dem der Quellcode gespeichert wird, und dann wird der Ausgabe-Bucket verschoben, um die AWS CodeBuild Ausgabe (konvertierten Code) aufzunehmen. Weitere Informationen finden Sie unter Erstellen, Konfigurieren und Arbeiten mit Amazon S3 S3-Buckets im Amazon S3 S3-Benutzerhandbuch.

1. Um den Projekt-Bucket zu erstellen, melden Sie sich bei der Amazon S3 S3-Konsole an und wählen Sie Create Bucket aus.
2. Geben Sie in der allgemeinen Konfiguration einen Namen für den Bucket ein und geben Sie an AWS-Region , wo Sie den Bucket erstellen möchten. Ein Beispiel für einen Namen ist*`codebuild-regionId-accountId-bucket`*, wobei:

   - `regionId`ist der AWS-Region des Buckets.
   - `accountId`ist deine 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.

Welche Namen Sie auch immer für diese Buckets wählen, achten Sie darauf, sie in diesem Tutorial zu verwenden.

## Schritt 3: IAM-Richtlinie erstellen

In diesem Schritt erstellen Sie eine IAM-Richtlinie. Die bereitgestellte IAM-Richtlinie gewährt spezifische Berechtigungen AWS CodeBuild für die Interaktion mit Amazon S3, Amazon Elastic Container Registry, CodeBuild generierten CloudWatch Amazon-Protokollen und Amazon Elastic Compute Cloud Ressourcen für die Codekonvertierung. Diese Richtlinie ist nicht auf Kunden zugeschnitten. Die Richtlinie gewährt Berechtigungen zur Interaktion und AWS Mainframe Modernization zum Abrufen der Statistiken zur Code-Konvertierung, um dem Kunden eine angemessene Rechnung zu stellen.

Weitere Informationen zum Erstellen einer IAM-Richtlinie finden Sie unter Erstellen von IAM-Richtlinien im IAM-Benutzerhandbuch.

So erstellen Sie eine Richtlinie

1. Melden Sie sich bei der IAM-Konsole an und wählen Sie im linken Navigationsbereich Richtlinien aus.
2. Wählen Sie Richtlinie erstellen aus.
3. Kopieren Sie die folgende JSON-Richtlinie und fügen Sie sie in den Richtlinien-Editor ein.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObjectAcl",
                "s3:GetBucketAcl"
            ],
            "Resource": [
                "arn:aws:s3:::codebuild-regionId-accountId-bucket",
                "arn:aws:s3:::codebuild-regionId-accountId-bucket/*",
                "arn:aws:s3:::aws-m2-repo-*" ],
                    "Effect": "Allow"
          },
          {
              "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "logs:*",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeDhcpOptions",
                "ec2:DescribeVpcs",
```

```
                "ec2:CreateNetworkInterfacePermission"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

4. Sie können der Richtlinie optional Tags hinzufügen. Bei Tags handelt es sich um Schlüssel-Wert-Paare, die Ihnen helfen können, den Zugriff auf die Richtlinie zu organisieren, nachzuverfolgen oder zu kontrollieren.
5. Klicken Sie auf Next:Review (Weiter: Prüfen).
6. Geben Sie einen Namen für die Richtlinie ein, z. B. *`CodeBuildAWSM2CCMPolicy`*
7. Sie können optional eine Beschreibung für die Richtlinie eingeben und die Richtlinienübersicht überprüfen, um sicherzustellen, dass sie korrekt ist.
8. Wählen Sie Richtlinie erstellen aus.

## Schritt 4: 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 Weiter aus.
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, `IAMRoleTaskExecutionRoleForCodeBuild` 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 5: Hängen Sie die IAM-Richtlinie an die IAM-Rolle an

In diesem Schritt fügen Sie die IAM-Richtlinie, die Sie zuvor erstellt haben, der `IAMRoleTaskExecutionRoleForCodeBuild` 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, `IAMRoleTaskExecutionRoleForCodeBuild` 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, *`CodeBuildAWSM2CCMPolicy`* z. B.
5. Wählen Sie Richtlinien anfügen.

## Schritt 6: Erstellen Sie das CodeBuild Projekt

In diesem Schritt erstellen Sie drei verschiedene CodeBuild Projekte auf der Grundlage der oben genannten `buildspec.yml` Datei.

## Schritt 6.1: Erstellen Sie das Define-Projekt

Um das Define-Projekt zu erstellen

1. Melden Sie sich bei der CodeBuild Konsole an und wählen Sie Create build project aus.
2. Geben Sie im Abschnitt Projektkonfiguration einen Namen für das Projekt ein, `1-awsm2ccm-define-project` z. B.
3. Behalten Sie im Abschnitt Quelle für Quellanbieter die Standardauswahl bei.
4. Wählen Sie im Abschnitt Umgebung die Option Benutzerdefiniertes Bild aus.
5. Wählen Sie im Feld Umgebungstyp die Option Linux aus.
6. Wählen Sie unter Image-Registrierung die Option Andere Registrierung aus.
7. Folgen Sie im Feld Externe Registrierungs-URL dem the section called “AWS Mainframe Modernization Container” Abschnitt.
8. Wählen Sie unter Servicerolle die Option Bestehende Servicerolle aus und wählen Sie im Feld Rollen-ARN die Servicerolle aus, die Sie zuvor erstellt haben (z. B.`IAMRoleTaskExecutionRoleForCodeBuild`).
9. Erweitern Sie den Abschnitt Zusätzliche Konfiguration und gehen Sie wie folgt vor:
   a. VPC: Konfigurieren Sie bei Bedarf basierend auf Ihrem Setup.
   b. Timeout: Auf 60 Minuten eingestellt.
   c. Timeout in der Warteschlange: Auf 480 Minuten eingestellt.
   d. Verschlüsselung: Wählen Sie die entsprechenden Verschlüsselungseinstellungen (Standard ist in Ordnung).
   e. Fügen Sie im Abschnitt Umgebungsvariablen nacheinander Folgendes hinzu:
      - Name: PROJECT_BUCKET. Wert:. **`codebuild-regionId-accountId- bucket`** Typ: Klartext
      - Name: PROJECT_DIR. Wert:. **`prj_codebuild_01`** Typ: Klartext
      - Name: AWSM2CCM_ACTION. Wert:. **`define_project`** Typ: Klartext
      - Name: AWSM2CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Wert:. Typ: Klartext
10. Wählen Sie im Abschnitt Buildspec die Option Build-Befehle einfügen und dann Zum Editor wechseln aus.
11. Ersetzen Sie die aktuellen Werte durch Folgendes:

```
version: 0.2
phases:
    build:
        commands:
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '**/*'
    discard-paths: no
    base-directory: $PROJECT_DIR
```

wobei PROJECT_DIR Umgebungsvariablen sind, die innerhalb verfügbar sind. CodeBuild Weitere Informationen finden Sie unter Umgebungsvariablen in Build-Umgebungen.

12. Gehen Sie im Abschnitt Artefakte wie folgt vor:

    - Wählen Sie unter Typ die Option Amazon S3 und dann Ihren Ausgabe-Bucket aus, zum Beispiel`codebuild-regionId-accountId-bucket`.
    - Lassen Sie dieses Feld für Pfad leer.
    - Geben Sie für Name ein**`prj_codebuild_01`**.
    - Wählen Sie für Artefaktverpackungen die Option Keine aus.
    - deaktivieren Sie diese Option für „Artefaktname überschreiben“.
    - Belassen Sie für Verschlüsselung die Standardeinstellungen.

13. Gehen Sie für den Abschnitt Protokolle wie folgt vor:

    - CloudWatch Protokolle: Deaktiviert
    - S3-Protokolle: Aktiviert
    - Bucket: **`codebuild-regionId-account-bucket`**
    - Protokollpfad: **`CODEBUILD-LOGS`**

14. Wählen Sie Create build project (Build-Projekt erstellen) aus.

## Schritt 6.2: Erstellen Sie das Codeanalyse-Projekt

Um das Codeanalyse-Projekt zu erstellen

1. Melden Sie sich bei der CodeBuild Konsole an und wählen Sie Build-Projekt erstellen aus.
2. Geben Sie im Abschnitt Projektkonfiguration einen Namen für das Projekt ein, `2-awsm2ccm-analysis` z. B.
3. Wählen Sie im Abschnitt Quelle für Quellanbieter Amazon S3 und dann den Eingabe-Bucket aus, den Sie zuvor erstellt haben (z. B.`codebuild-regionId-accountId-bucket`).
4. Geben Sie im Feld S3-Objektschlüssel oder S3-Ordner Folgendes ein**`prj_codebuild_01`**.
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. Folgen Sie im Feld Externe Registrierungs-URL dem the section called “AWS Mainframe Modernization Container” Abschnitt.
9. Wählen Sie unter Servicerolle die Option Bestehende Servicerolle aus und wählen Sie im Feld Rollen-ARN die Servicerolle aus, die Sie zuvor erstellt haben (z. B.`IAMRoleTaskExecutionRoleForCodeBuild`).
10. Erweitern Sie den Abschnitt Zusätzliche Konfiguration und gehen Sie wie folgt vor:
    a. VPC: Konfigurieren Sie bei Bedarf basierend auf Ihrem Setup.
    b. Timeout: Auf 60 Minuten eingestellt.
    c. Timeout in der Warteschlange: Auf 480 Minuten eingestellt.
    d. Verschlüsselung: Wählen Sie die entsprechenden Verschlüsselungseinstellungen (Standard ist in Ordnung).
    e. Fügen Sie im Abschnitt Umgebungsvariablen nacheinander Folgendes hinzu:
       - Name: PROJECT_BUCKET. Wert:. **`codebuild-regionId-accountId-bucket`** Typ: Klartext
       - Name: PROJECT_DIR. Wert:. **`prj_codebuild_01`** Typ: Klartext
       - Name: AWSM2CCM_ACTION. Wert:. **`analysis`** Typ: Klartext
       - Name: AWSM2CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Wert:. Typ: Klartext

11. Wählen Sie im Abschnitt Buildspec die Option Build-Befehle einfügen und dann Zum Editor wechseln aus.
12. Ersetzen Sie die aktuellen Werte durch Folgendes:

```
version: 0.2
phases:
    build:
        commands:
            - ln -s $CODEBUILD_SRC_DIR $PROJECT_DIR
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '*.log'
        - '_Converted/*/*'
        - '_Reports/*'
    secondary-artifacts:
        reports:
            files:
                - '_Reports/AWSM2CCM*'
    discard-paths: no
    base-directory: $PROJECT_DIR
```

    wobei PROJECT_DIR Umgebungsvariablen sind, die innerhalb verfügbar sind. CodeBuild Weitere Informationen finden Sie unter Umgebungsvariablen in Build-Umgebungen.
13. Gehen Sie im Abschnitt Artefakte wie folgt vor:
    - Wählen Sie unter Typ die Option Amazon S3 und dann Ihren Ausgabe-Bucket aus (z. B.`codebuild-regionId-accountId-bucket`).
    - Geben Sie als Pfad ARTIFACTS ein.
    - Geben Sie als Namen ein**`prj_codebuild_01`**.
    - Wählen Sie für Artefaktverpackungen die Option Keine aus.
    - deaktivieren Sie diese Option für „Artefaktname überschreiben“.
    - Belassen Sie für Verschlüsselung die Standardeinstellungen.
14. Gehen Sie für den Abschnitt Protokolle wie folgt vor:
    - CloudWatch Protokolle: Deaktiviert
    - S3-Protokolle: Aktiviert

- Bucket: **`codebuild-regionId-account-bucket`**
- Protokollpfad: **`CODEBUILD-LOGS`**

15. Wählen Sie Create build project (Build-Projekt erstellen) aus.

## Schritt 6.3: Erstellen Sie das Code-Konvertierungsprojekt

Um das Code-Konvertierungsprojekt zu erstellen

1. Melden Sie sich bei der CodeBuild Konsole an und wählen Sie Build-Projekt erstellen aus.
2. Geben Sie im Abschnitt Projektkonfiguration einen Namen für das Projekt ein (z. B.`3-awsm2ccm-convert`).
3. Wählen Sie im Abschnitt Quelle für Quellanbieter Amazon S3 und dann den Eingabe-Bucket aus, den Sie zuvor erstellt haben (z. B.`codebuild-regionId-accountId-bucket`).
4. Geben Sie im Feld S3-Objektschlüssel oder S3-Ordner Folgendes ein**`prj_codebuild_01`**.
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. Folgen Sie im Feld Externe Registrierungs-URL dem the section called “AWS Mainframe Modernization Container” Abschnitt.
9. Wählen Sie unter Servicerolle die Option Existierende Servicerolle aus und wählen Sie im Feld Role ARN die Servicerolle aus, die Sie zuvor erstellt haben, `IAMRoleTaskExecutionRoleForCodeBuild` z. B.
10. Erweitern Sie den Abschnitt Zusätzliche Konfiguration und gehen Sie wie folgt vor:

    a. VPC: Konfigurieren Sie bei Bedarf basierend auf Ihrem Setup.

    b. Timeout: Auf 60 Minuten eingestellt.

    c. Timeout in der Warteschlange: Auf 480 Minuten eingestellt.

    d. Verschlüsselung: Wählen Sie die entsprechenden Verschlüsselungseinstellungen (Standard ist in Ordnung).

    e. Fügen Sie im Abschnitt Umgebungsvariablen nacheinander Folgendes hinzu:

    - Name: PROJECT_BUCKET. Wert:. **`codebuild-regionId-accountId-bucket`** Typ: Klartext

- Name: PROJECT_DIR. Wert:. **`prj_codebuild_01`** Typ: Klartext
- Name: AWSM2CCM_ACTION. Wert:. **`conversion`** Typ: Klartext
- Name: AWSM2CCM_LOGGING_BUCKET. **`s3:// codebuild-regionId-accountId-bucket`**Wert:. Typ: Klartext

11. Wählen Sie im Abschnitt Buildspec die Option Build-Befehle einfügen und dann Zum Editor wechseln aus.
12. Ersetzen Sie die aktuellen Werte durch Folgendes:

```
version: 0.2
phases:
    build:
        commands:
            - export AWSM2CCM_PUSH_RUNTIME_COPYBOOKS=y
            - ln -s $CODEBUILD_SRC_DIR $PROJECT_DIR
            - . /app/awsm2ccm_prod/bin/setup_env.sh
            - run_awsm2ccm.sh $PROJECT_DIR
artifacts:
    files:
        - '*.log'
        - '_Converted/*/*'
        - '_Reports/*'
    discard-paths: no
    base-directory: $PROJECT_DIR
```

wobei PROJECT_DIR Umgebungsvariablen sind, die innerhalb verfügbar sind. CodeBuild Weitere Informationen finden Sie unter Umgebungsvariablen in Build-Umgebungen.

13. Gehen Sie im Abschnitt Artefakte wie folgt vor:
    - Wählen Sie unter Typ die Option Amazon S3 und dann Ihren Ausgabe-Bucket aus (z. B.`codebuild-regionId-accountId-bucket`).
    - Geben Sie als Pfad ARTIFACTS ein.
    - Geben Sie als Namen ein**`prj_codebuild_01`**.
    - Wählen Sie für Artefaktverpackungen die Option Keine aus.
    - deaktivieren Sie diese Option für „Artefaktname überschreiben“.
    - Belassen Sie für Verschlüsselung die Standardeinstellungen.
14. Gehen Sie für den Abschnitt Protokolle wie folgt vor:

- CloudWatch Protokolle: Deaktiviert
- S3-Protokolle: Aktiviert

- Bucket: **`codebuild-regionId-account-bucket`**
- Protokollpfad: **`CODEBUILD-LOGS`**

15. Wählen Sie Create build project (Build-Projekt erstellen) aus.

## Schritt 7: Definieren Sie das Projekt und laden Sie den Quellcode hoch

Das Define Project richtet den Projektordner und die Konfigurationsdateien ein, die mit Standardkonfigurationen initialisiert wurden. In diesem Schritt starten Sie den Build. So gehen Sie vor:

1. Loggen Sie sich in die AWS CodeBuild Konsole ein.
2. Wählen Sie im linken Navigationsbereich Projekte erstellen aus.
3. Wählen Sie das zuvor erstellte Projekt (`1-awsm2ccm-define-project`) aus, das erstellt werden soll
4. Wählen Sie Build starten und dann Jetzt starten, um das Projekt zu definieren. Sobald der Build gestartet wird, ändert sich der Status in In Bearbeitung.
5. Wählen Sie „Phasendetails“, um den Fortschritt der einzelnen Schritte zu sehen, die vom AWS CodeBuild Projekt orchestriert werden.
6. Warten Sie, bis sich der Status für alle Schritte auf erfolgreich geändert hat.
7. Gehen Sie zur Amazon S3 S3-Konsole.
8. Suchen Sie den Amazon S3 S3-Bucket mit dem Namen und klicken Sie darauf `codebuild-regionId-accountId-bucket`

   - **`CODEBUILD-LOGS/`**Der Ordner enthält die AWS CodeBuild Protokolle für die laufenden AWS CodeBuild Projekte.
   - **`prj_codebuild_01/`**Ordner, der die Projektstruktur enthält. Er wird in den Schritten Analyse, expand_macros und Convert verwendet. Sie können wählen`prj_codebuild_01/`, ob Sie Details untersuchen möchten
   - **`cobol_reserved.rsw`**Konfigurationsdatei (Liste der COBOL-Wörter), die für den Konverter reserviert ist. Sie wird während des Konvertierungsschritts verwendet.

- **`Macro_Expansion/`**Der Ordner enthält Makros, die in Assembler-Programme erweitert werden können. Er wird während des Schritts expand_macros verwendet.
- **`macro_settings.json`**Die Konfigurationsdatei enthält eine benutzerdefinierte Makro-Ersetzung. Sie wird während des Schritts expand_macros verwendet.
- **`macrolib/`**Der Ordner enthält die zu konvertierenden Assembler-Makros. Er wird während des Analyse- und Konvertierungsschritts verwendet.
  1. Wählen Sie `macrolib/`.
  2. Standardmäßig wird ein Assembler-Makro mit dem Namen `MACRO1.mac` als Beispieldatei bereitgestellt. Löschen Sie diese Datei, da sie für die Analyse nicht benötigt wird.
  3. Laden Sie Ihre Makros in dieses Verzeichnis hoch.
- **`project_settings_aux.json`**Die Konfigurationsdatei enthält Einstellungen, die sich auf die Codepage beziehen. Sie wird während des Konvertierungsschritts verwendet.
- **`project_settings.json`**Die Konfigurationsdatei enthält Einstellungen für den Konverter. Sie wird während des Konvertierungsschritts verwendet.
- **`srclib/`**Der Ordner enthält die zu konvertierenden Assembler-Programme. Er wird während des Analyse- und Konvertierungsschritts verwendet.
  1. Wählen Sie `srclib/`.
  2. Standardmäßig sind zwei Assembler-Programme benannt `SQtest01.asm` und `SQtest02.asm` werden als Beispiele bereitgestellt. Löschen Sie diese Dateien, da sie für Ihre Analyse und Konvertierung nicht benötigt werden.
  3. Laden Sie Ihre Assembler-Programme in dieses Verzeichnis hoch.

9. Überprüfen Sie den Status für `1-awsm2ccm-define-project` Schritt. Der Vorgang sollte auf der Registerkarte Neuester Build-Status erfolgreich gewesen sein.

Sie sind bereit für den nächsten Schritt: die Codeanalyse.

## Schritt 8: Führen Sie die Analyse durch und verstehen Sie die Berichte

> **Note**
>
> AWS Mainframe Modernization Der Analyseschritt zur Codekonvertierung ist kostenlos.

In diesem Schritt starten Sie einen weiteren Build:

1. Wählen Sie im linken Navigationsbereich die Option Projekte erstellen aus.
2. Wählen Sie das Projekt aus, das Sie in Schritt 6.2 erstellt haben, um Folgendes zu erstellen:`2-awsm2ccm-analysis`.
3. Wählen Sie Build starten und dann Jetzt starten aus, um Analyseberichte zu generieren. Dadurch wird der Build gestartet und der Status wird auf In Bearbeitung geändert.
4. Wählen Sie „Phasendetails“ aus, in denen Sie den Fortschritt der einzelnen vom AWS CodeBuild Projekt orchestrierten Schritte sehen können. Warten Sie, bis sich der Status für alle Schritte auf erfolgreich ändert.
5. Gehen Sie von der AWS Management Console zur Amazon S3 S3-Servicekonsole.
6. Suchen Sie den Amazon S3 S3-Bucket und klicken Sie darauf: `codebuild-regionId-accountId-bucket`

    a. **`ARTIFACTS/`**Der Ordner enthält die Ergebnisse der Analyse - und Konvertierungsschritte.

    b. Wählen Sie `ARTIFACTS/prj_codebuild_01/_Reports/`.

    c. Die folgenden Berichte werden verfügbar sein:

    - `AWSM2CCM-Analysis-Report-<timestamp>.pdf`ist ein Bericht für Führungskräfte, der Aufschluss über die Abrechnung und den Umfang der AWS Mainframe Modernization Code-Konvertierung gibt. Außerdem verbessert er die Konversionsrate, die Konversionsübersicht und die detaillierten Konversionsstatistiken. Er fasst auch die Anzahl der Codes und die Anzahl der abrechnungsfähigen Codes auf Projektebene zusammen und enthält Kennzahlen und Listen der referenzierten Mitglieder für jede Komponente. Es ist wichtig, diesen Bericht vor der eigentlichen Konvertierung auszuführen und zu prüfen.
    - `Conversion_Detailed_Statistics.txt`enthält die Häufigkeit und das erwartete Konvertierungsergebnis (angezeigt als „Konvertierungsstatus“) für jede Anweisung in jeder Komponente. Auf diese Weise kann schnell festgestellt werden, ob Anweisungen eindeutig sind, die der Konverter nicht unterstützt. Mögliche Ergebnisse zum Konvertierungsstatus sind:
        - Vollständig konvertiert: Die Anweisung wird korrekt in COBOL konvertiert.
        - Teilweise konvertiert: Die Anweisung wird unterstützt, verwendet jedoch einen Parameter oder Ausdruck, der nicht unterstützt wird. Nach der Konvertierung sind wahrscheinlich manuelle Anpassungen erforderlich.
        - Nicht konvertiert: Die Anweisung wird vom Konverter nicht unterstützt.

- Anweisungen zur Überprüfung vor der Kompilierung: Diese sind normalerweise in den Makros enthalten und beziehen sich auf Befehle, die wahrscheinlich auch als Conditional Assembly Language (z. B. AIF, AGO) auf dem Mainframe bekannt sind. Diese werden vom Pre-Compiler verarbeitet, der von solchen Instruktionen oder Direktiven gesteuert wird, sauberen/statischen ASM-Code auswählt und erzeugt. Diese Anweisungen hängen von den tatsächlichen Werten der Makroparameter ab, die kompiliert werden. Daher kann dasselbe Makro je nach den Werten der übergebenen Parameter unterschiedliche Teile von ASM-Code generieren. Dies liegt an dem Vorhandensein solcher Anweisungen vor der Kompilierung. In diesem Fall sollten Sie erwägen, das Makro zu erweitern oder neu zu entwickeln.

- `Conversion_Global_Statistics.txt`bietet eine Zusammenfassung des Konvertierungsstatus auf Komponentenebene.
- `CrossReference_PgmToCpyMacro.txt`berichtet über die Abhängigkeiten von Assembler-Programmen von Makros. Es bietet eine schnelle Methode, um festzustellen, ob Makros im hochgeladenen Code fehlen.
- `CrossReference_PgmToPgm.txt`berichtet über Abhängigkeiten von Assembler-Programmen von anderen Assembler-Programmen. Es bietet eine schnelle Methode, um festzustellen, ob Assembler-Programme im hochgeladenen Code fehlen.

7. Kehren Sie zur AWS CodeBuild Servicekonsole zurück.
8. Überprüfen Sie den Status für den 2-awsm2ccm-Analyseschritt. Der Vorgang sollte auf der Registerkarte Neuester Build-Status erfolgreich gewesen sein.

Sie sind bereit für den nächsten Schritt: Codekonvertierung.

## Schritt 9: Führen Sie die Codekonvertierung aus

> **Important**
>
> AWS Mainframe Modernization Der Konvertierungsschritt für die Codekonvertierung wird je nach Nutzung in Rechnung gestellt. Weitere Informationen zur Fakturierung finden Sie unter the section called “Verstehen Sie die Abrechnung der Codekonvertierung”.

In diesem Schritt konfigurieren Sie den Konvertierungsprozess und starten dann den Build.

1. Gehen Sie von der AWS Management Console zum Amazon S3 S3-Service.

2. Suchen Sie den Amazon S3 S3-Bucket und klicken Sie darauf:`codebuild-regionId-accountId-bucket`.

   a. Wechseln Sie zu `prj_codebuild_01/`.

   b. Wählen Sie `project_settings.json` und wählen Sie Herunterladen.

   c. Öffnen Sie die `project_settings.json` Datei, um die folgende JSON-Struktur zu sehen:

```
{
"Source programs directory":"srclib",
"Source copybooks/macros directory":"macrolib",
"Copybook/Macros Conversion":"Called_only",
"Do not regenerate the Copy/Macro if already exists":"false",
"Target Compiler":"IBM",
"Endianess":"Big",
"Converted programs extension":"",
"Converted CICS programs extension":"",
"Converted copies/macros extension":"",
"Trace Level":"STANDARD",
"Trace file open mode":"append",
"Data definition level":5,
"Start picture column":40,
"Generate Sync FILLER with name":"FILL-SYNC",
"Use SYNC clause":"yes",
"Decimal Point Comma":"true",
"Original Source Placement":"RIGHT"
}
```

   wobei

   - Quellprogrammverzeichnis: enthält die Assembler-Programme, die für die Konvertierung benötigt werden.
   - Quellverzeichnis CopyBooks/Macros: enthält die Assembler-Makros und Copybooks, die für die Konvertierung benötigt werden.
   - Die Konvertierung von Copybooks/Macros kann entweder erfolgen:
     - Alle: Dieses Optionsfeld gibt an, dass bei der vollständigen Konvertierung alle im Verzeichnis copybook/Macros verfügbaren Dateien konvertiert werden, unabhängig davon, ob das Verzeichnis von den Programmen verwendet wird oder nicht.

- Called_only: Dieses Optionsfeld gibt an, dass bei der vollständigen Konvertierung nur das konvertiert wird, was tatsächlich von den copybook/Macros Programmen verwendet wird.

- > **Important**
  >
  > Sie müssen das nicht neu generieren, wenn es bereits vorhanden ist Copy/Macro .

  Wenn dies zutrifft, konvertiert das Tool das CopyBook/das Makro nicht erneut, wenn es bereits konvertiert wurde (im Ausgabeordner vorhanden ist).

- Ziel: Die Konvertierung der Programme (generierter Code) hängt vom Ziel-COBOL-Compiler ab. Die folgenden Optionen werden unterstützt:
  - „IBM“ für IBM Mainframe
  - „MF“ für Micro Focus COBOL
  - „VERYANT“ für Veryant ist COBOL
  - „NTT“ für NTT DATA Enterprise COBOL (Unikix)
- Endianess und Bitness: Die Konvertierung der Programme (generierter Code) hängt von der Zielplattform ab (Bit/Endianess). Diese Kombination ermöglicht die Auswahl der folgenden unterstützten Optionen:
  - Endianess: Big (für Big-Endian)/Little (Little-Endian). Zum Beispiel ist IBM z/OS Mainframe Big-Endian, Windows ist Little-Endian, Linux ist je nach Distribution unterschiedlich (z. B. ist Amazon Linux 2 auf Little-Endian). EC2
  - Bitanzahl: 32/64 (wenn nicht angegeben, ist der Standardwert 32). Die empfohlene Einstellung ist 32 Bit.
- Konvertierte Programmerweiterung: Damit wird die Dateierweiterung für die generierten COBOL-Programme festgelegt. Leer („“): keine Erweiterung. Für COBOL-Ziele von Rocket Software (ehemals Micro Focus) wird CBL empfohlen, damit Rocket Enterprise Developer die Dateien korrekt erkennen kann.
- Konvertierte CICS-Programmerweiterung: Damit wird die Dateierweiterung für die generierten CICS-COBOL-Programme festgelegt. Leer („“): keine Erweiterung. Für COBOL-Ziele von Rocket Software wird CBL empfohlen, damit Rocket Enterprise Developer die Dateien korrekt erkennen kann.

- Erweiterung „Konvertierte Copybooks/Macros“: Damit wird die Dateierweiterung für die generierten COBOL-Copybooks festgelegt. Leer („“): keine Erweiterung. Für COBOL-Ziele von Rocket Software wird CPY empfohlen, damit Rocket Enterprise Developer die Dateien korrekt erkennen kann.
- Trace-Level: Trace ist die Information, die CodeBuild während der Konvertierung protokolliert wird. Der Benutzer kann den Detaillierungsgrad auswählen, indem er eine der bereitgestellten Optionen auswählt.
  - ERROR = TRACE ERROR: Es werden nur Konvertierungsfehler angezeigt.
  - STANDARD = TRACE STANDARD: Konvertierungsfehler und Standardinformationen werden angezeigt. Dies ist die empfohlene Einstellung.
  - ALL = TRACE ALL: maximale Ablaufverfolgungsstufe
- Modus zum Öffnen der Trace-Datei: Wird nicht verwendet. Die Standardeinstellung „Append“ wird empfohlen.
- Datendefinitionsebene: Dies gibt die Anfangsebene der Unterfelder (nach der Ebene „01“) an, die im Bereich Working-Storage und Linkage definiert sind. Muss eine Zahl sein.
- Spalte mit dem Startbild: Hier geht es um das Format des generierten COBOL-Codes und gibt die Spalte an, in der die PIC-Klausel steht (nach den Feldnamen). Muss eine Zahl sein.
- Platzierung der ursprünglichen Quelle: Dies gibt die Position an, an der die Kommentare im Programm platziert wurden. Es gibt zwei Optionen:
  - RECHTS: Diese Option platziert den Kommentar oder die zusätzlichen Informationen an der richtigen Position nach der dreiundsiebzigsten (73) Spalte. In COBOL wird der Code in die ersten zweiundsiebzig (1—72) Spalten geschrieben, und alles aus der dreiundsiebzigsten (**>=** 73) Spalte wird als Kommentar behandelt.
  - OBEN: Mit dieser Option wird der Kommentar über dem übersetzten Inhalt platziert.
- Sync FILLER mit Namen generieren: Diese Option bezieht sich auf die Ausrichtung von Binärfeldern im Speicher (Assembler-Datentypen „H“, „F“, „D“, die in den COBOL-Datentyp „COMP“ konvertiert werden). Um die korrekte Ausrichtungsgrenze zu gewährleisten, werden während der Konvertierung explizite Füllfelder hinzugefügt. Dies ist eine textbasierte Option, der Wert muss eine Zeichenfolge sein (wie FILL-SYNC).
- SYNC-Klausel verwenden: Diese Option bezieht sich auf die Ausrichtung von Binärfeldern im Speicher. Ja = alle Felder wurden in COBOL konvertiert. „COMP“ wird mit der Klausel „SYNC“ definiert (z. B. 05 WRKFLD PIC S9 (09) COMP SYNC).

- Dezimalkomma: Wenn dies zutrifft, wird die DECIMAL-POINT IS COMMA-Klausel zum COBOL-Absatz „SPECIAL-NAMES“ hinzugefügt.

d. Ändern Sie entsprechend Ihren Anforderungen die entsprechenden Parameter und speichern Sie dann die. `project_settings.json`

e. Entfernen Sie die vorhandene `project_settings.json` Datei aus `prj_codebuild_01/` dem Amazon S3 S3-Bucket und laden Sie dann die neue Version hoch.

3. Gehen Sie zurück zum AWS CodeBuild Service.
4. Wählen Sie das Projekt aus, das Sie zuvor erstellt haben: `3-awsm2ccm-convert`

a. Wählen Sie Build starten und dann Jetzt starten, um Assembler-Programme und Makros in COBOL-Programme und Copybooks zu konvertieren.

b. Warten Sie, bis sich der Build-Status für dieses Projekt auf Erfolgreich geändert hat. Es wird auf der Registerkarte Neuester Build-Status angezeigt.

## Schritt 10: Überprüfen Sie die Codekonvertierung

1. Gehen Sie vom AWS Management Console zum Amazon S3 S3-Service.
2. Suchen Sie den Amazon S3 S3-Bucket und klicken Sie darauf:`codebuild-regionId-accountId-bucket`.
3. Navigieren Sie zu **`awsm2ccm-do-not-delete`** . AWS Mainframe Modernization Bei der Codekonvertierung werden während des Konvertierungsvorgangs kodierte Binärdateien für jedes Assembler- oder Macro-Modul erstellt. Diese Dateien sind wichtig, um doppelte Rechnungen an die Kunden zu vermeiden und um nachzuverfolgen, wie viel des bereitgestellten Assembler-Codes analysiert und konvertiert wurde. Dateien werden am folgenden Ort gespeichert:`codebuild-regionId-accountId- bucket/awsm2ccm-do-not-delete/<your_AWS_account_id>/Hash`. Die codierten Dateien enthalten keinen Assembler-Code und es ist auch nicht möglich, Kundencode aus diesen Dateien zu extrahieren.

> **Important**
>
> Sie können diese Dateien auch nicht manuell bearbeiten oder löschen. Das Bearbeiten oder Löschen dieser Dateien kann zu mehreren Fakturierungen für dieselben Komponenten führen.

Behandeln Sie den **awsm2ccm-do-not-delete/**Ordner wie ein vom System verwaltetes Verzeichnis. Konsultieren Sie, Support bevor Sie Änderungen an diesem Verzeichnis oder seinem Inhalt vornehmen.

4. Klicken Sie hier`codebuild-regionId-accountId-bucket`, um zum Bucket zurückzukehren.
5. Wähle **ARTIFACTS/prj_codebuild_01/**. Der Ordner _Converted/ enthält die als Ergebnis des Codekonvertierungsschritts generierten COBOL-Ausgaben. Er wird die folgenden Unterverzeichnisse haben:
   - Der Ordner copybooks/ enthält die generierten COBOL-Copybooks.
   - Der Ordner programs/ enthält die generierten COBOL-Programme.
   - Der Ordner runtime_lib/ enthält zusätzliche COBOL-Programme und Copybooks, die von der Lösung bereitgestellt werden.
6. Wenn aus den Analyseberichten und anderen Berichten hervorgeht, dass die Konvertierung erfolgreich war und das AWS CodeBuild Projekt als Erfolgreich markiert **3-awsm2ccm-convert** ist, laden Sie den COBOL-Code und die Copybooks aus dem Verzeichnis _Converted/ herunter.

## Schritt 11: Laden Sie den konvertierten Code herunter

Laden Sie in diesem Schritt den COBOL-Code und die Copybooks aus dem Verzeichnis _Converted/ herunter und kompilieren Sie sie in der COBOL-Zielumgebung.

1. Gehen Sie vom AWS Management Console zum Amazon S3 S3-Service.
2. Suchen Sie den Amazon S3 S3-Bucket und klicken Sie darauf:`codebuild-regionId-accountId-bucket`.
3. Navigieren Sie zum Standort: `ARTIFACTS/prj_codebuild_01/_Converted/`.
4. Laden Sie den konvertierten COBOL-Code aus allen Unterverzeichnissen unter _Converted/ herunter. Sie können sie auch mit dem folgenden CLI-Befehl auf einmal herunterladen:

```
aws s3 cp s3://codebuild-regionId-accountId-
bucket/ARTIFACTS/prj_codebuild_01/_Converted/ . --recursive
```

5. Analysieren und kompilieren Sie das konvertierte COBOL in der COBOL-Zielumgebung.

# 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 finden Sie unter Löschen von Rollen oder Instanzprofilen im IAM-Benutzerhandbuch.
- Löschen Sie das CodeBuild Projekt, das Sie für dieses Tutorial erstellt haben. Weitere Informationen finden Sie  CodeBuildim AWS CodeBuild Benutzerhandbuch unter Löschen eines Build-Projekts.

# 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 (vertreten durch die Sun SPARCstation 20): ursprünglich eine Sun-4-Variante mit mehreren Prozessoren, die auf dem in der 600MP-Serie eingeführten Prozessormodulbus basiert. MBus SPARCServer Die Sun-4M-Architektur umfasste später auch Systeme ohne MBus Einprozessorsysteme wie die SPAR Cstation 5, die Prozessoren mit SPARC V8-Architektur nutzten. Wird ab SunOS 4.1.2 und von Solaris 2.1 bis Solaris 9 unterstützt. SPARCServer Die Unterstützung für 600 MP wurde nach Solaris 2.5.1 eingestellt.

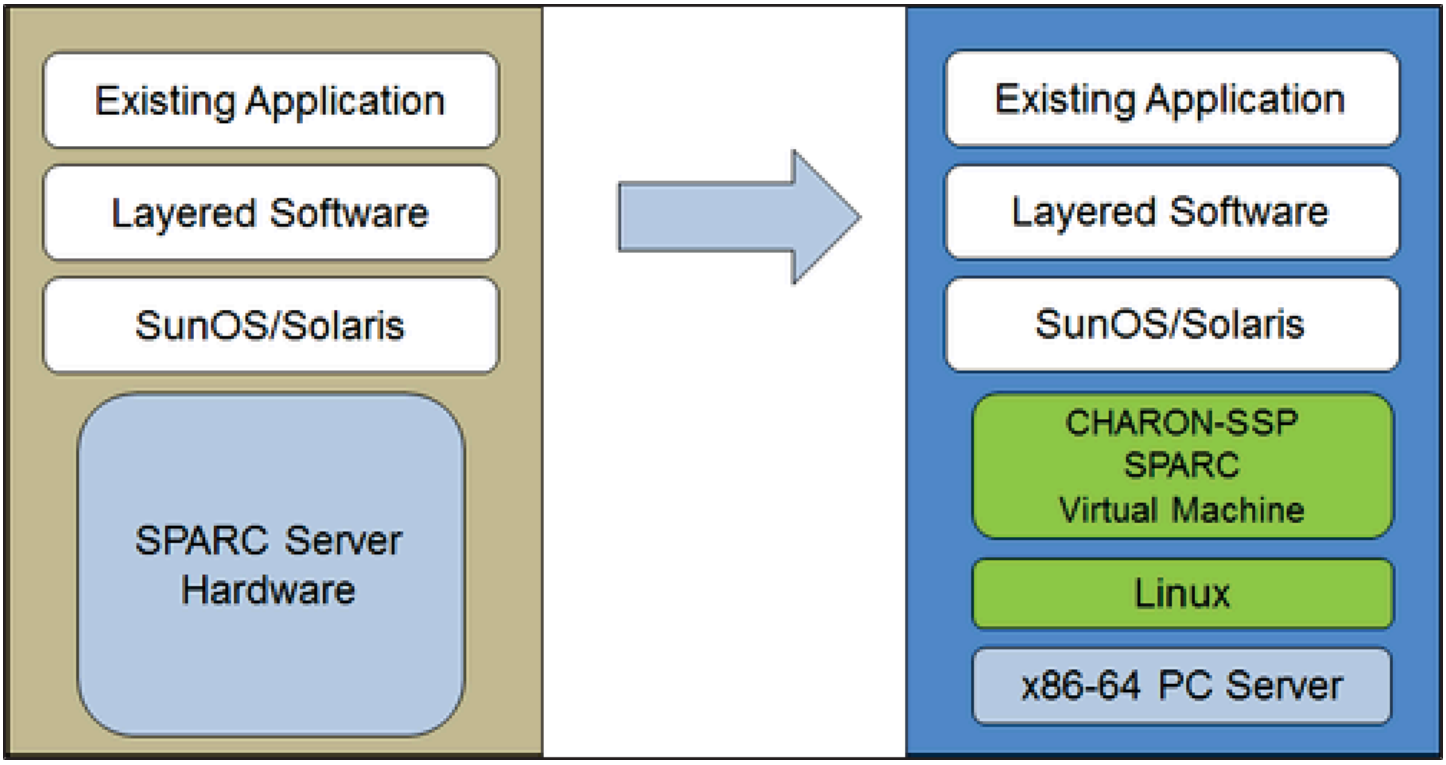
Charon-SSP/4U (+) emuliert die folgende SPARC-Hardware:

- 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 UltraSP ARC 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 ab Version 3/05 von Solaris Version 10 unterstützt HW2 (die meisten Modelle — einschließlich der von Charon-SSP emulierten Hardware — benötigen 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 /EPT von Intel und AMD AMD-V/NPT

in der heutigen Zeit, um eine bessere virtuelle CPU-Leistung zu bieten. VTx CPUs Charon-SSP/4U+ und Charon-SSP/4V+ erfordern VT-x/EPT- oder AMD-V/NPT-Unterstützung und müssen auf einem dedizierten Hostsystem installiert werden. CPUs Die Ausführung dieser Produktvarianten in einer VM (z. B. on) 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 CPUs ist ein System mit vielen emulierten Komponenten 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 CPU-Kernzuweisung für emulierte Kerne 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 Instance

Allgemeine CPU-Anforderungen: Charon-SSP unterstützt moderne Prozessoren mit der x86-64-Architektur, die auf Amazon-Instances basieren. EC2

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 für den Charon-Host sichtbaren Anzahl der für den Charon-Host CPUs sichtbaren Daten 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 Grafikemulation zu ermöglichen.
- Wenn Hyperthreading auf modernen x86-64-Betriebssystemen aktiviert ist, können zwei Threads auf einem physischen CPU-Kern ausgeführt werden CPUs, sodass zwei Threads für das Host-Betriebssystem logisch sind. CPUs Deaktivieren Sie nach Möglichkeit Hyperthreading auf dem Charon-SSP-Host. In Cloud-Umgebungen ist dies jedoch häufig nicht möglich, oder es ist unklar, ob Hyperthreading verwendet wird oder nicht. VMware 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-Instance einzurichten AWS, benötigen Sie ein AWS Konto mit Administratorzugriff.
  - Identifizieren Sie die AWS Region, in der Sie Ihre Instance starten möchten. Stellen Sie sicher, dass die AWS Dienste, die Sie nutzen möchten, in dieser Region verfügbar sind. Weitere Informationen finden Sie unter AWS Dienste nach Regionen.
  - Identifizieren Sie die VPC und das Subnetz, in denen Sie Ihre Instance starten möchten.
  - Wenn Ihre Instance Internetzugang benötigt, stellen Sie sicher, dass die mit Ihrer VPC verknüpfte Routing-Tabelle über ein Internet-Gateway verfügt. Wenn Ihre Instance 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 an AWS Management Console und öffnen Sie die EC2 Amazon-Konsole 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. 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 an. 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.

6. 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 auf 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.

7. Wählen Sie im Abschnitt Netzwerkeinstellungen die Option Bearbeiten aus. Wählen Sie die Einstellungen, 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.

8. 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 mehr Speicher hinzufügen. Die Größe der Systemfestplatte muss jedoch den Linux-

Systemanforderungen entsprechen, einschließlich aller Anwendungen und Dienstprogramme, die Sie installieren möchten.

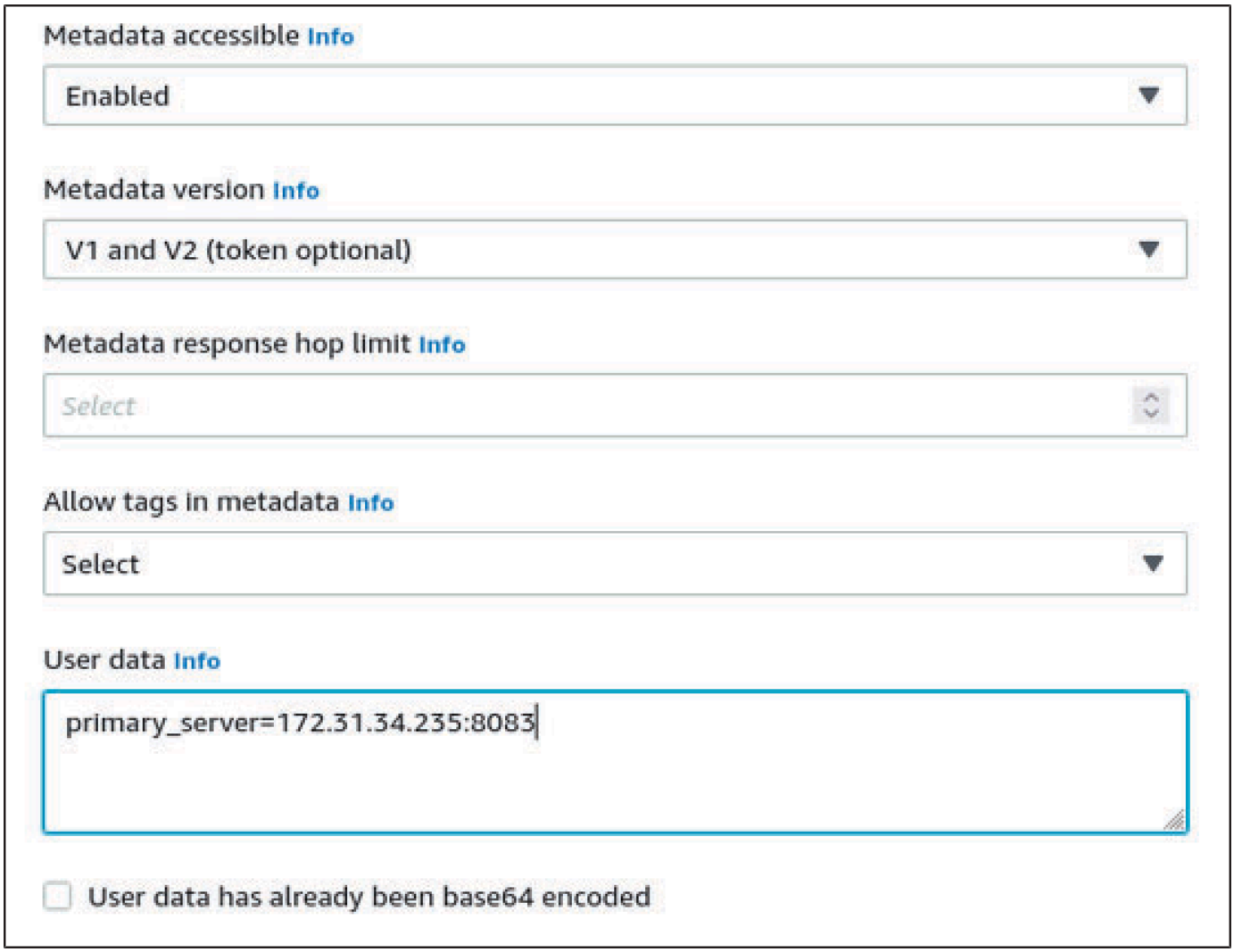
> **Note**
>
> Wir empfehlen, separate Speichervolumes für Charon-Anwendungsdaten (z. B. Festplatten-Images) zu erstellen. Bei Bedarf können Sie solche Volumes später auf eine andere Instanz migrieren.

9. 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.




10. Für ein 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.
11. 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 Instance-Konfiguration hinzufügen, bevor Sie die Instance starten.

Geben Sie die Informationen für den AutoVE-Lizenzserver ein, wie in der folgenden Abbildung gezeigt.




Die folgenden Optionen für die Konfiguration von Benutzerdaten sind gültig:

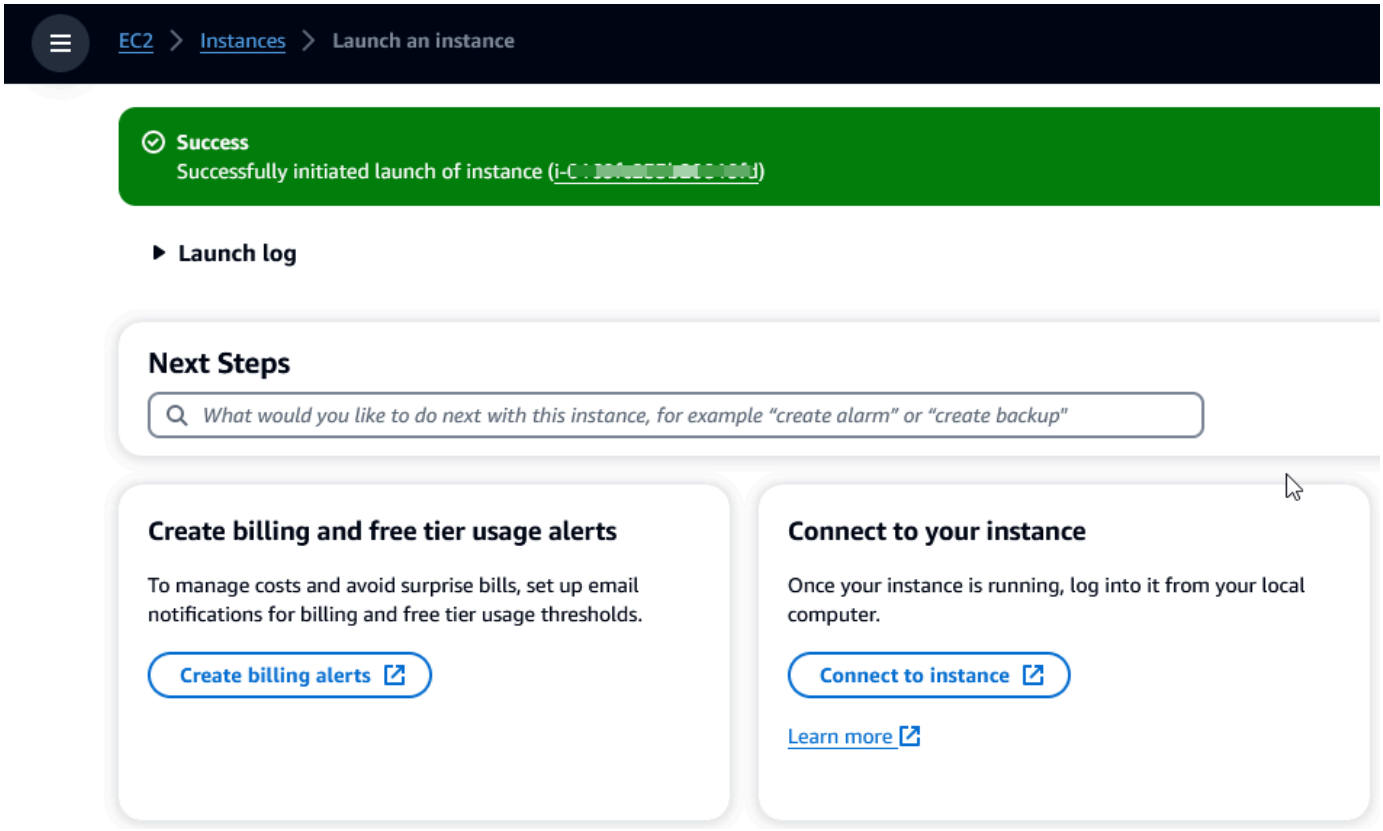
- **`primary_server=`***`<ip-address>`*`[:`*`<port>`*`]`
- **`backup_server=`***`<ip-address>`*`[:`*`<port>`*`]`

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 Ihnen die folgende Erfolgsmeldung angezeigt:




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 dazu, wie Sie eine Verbindung zu Ihrer Instance herstellen, finden Sie unter Connect im EC2 Amazon-Benutzerhandbuch.

# AWS Modernisierung von Mainframes und Neuplattformen mit NTT DATA

AWS Mainframe Modernization bietet eine Vielzahl von Amazon Machine Images (AMIs). Diese AMIs ermöglichen die schnelle Bereitstellung von EC2 Amazon-Instances und schaffen eine maßgeschneiderte Umgebung für das Rehosting und die Neuplattformierung von Mainframe-Anwendungen mithilfe von NTT Data. AWS Dieses Handbuch enthält die erforderlichen Schritte, um auf diese zuzugreifen und sie zu verwenden. AMIs

## Voraussetzungen

- Stellen Sie sicher, dass Sie Administratorzugriff auf ein AWS Konto haben, mit dem Sie EC2 Amazon-Instances erstellen können.
- Stellen Sie sicher, dass der AWS Mainframe-Modernisierungsservice in der Region verfügbar ist, in der Sie die EC2 Amazon-Instances erstellen möchten. Sehen Sie sich die Liste der verfügbaren AWS-Services nach Regionen an.
- Identifizieren Sie die Amazon VPC, in der Sie die EC2 Amazon-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 https://console.aws.amazon.com/marketplace an AWS Management Console und öffnen Sie die AWS Marketplace Konsole.
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-eg227ymldsnx2
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.

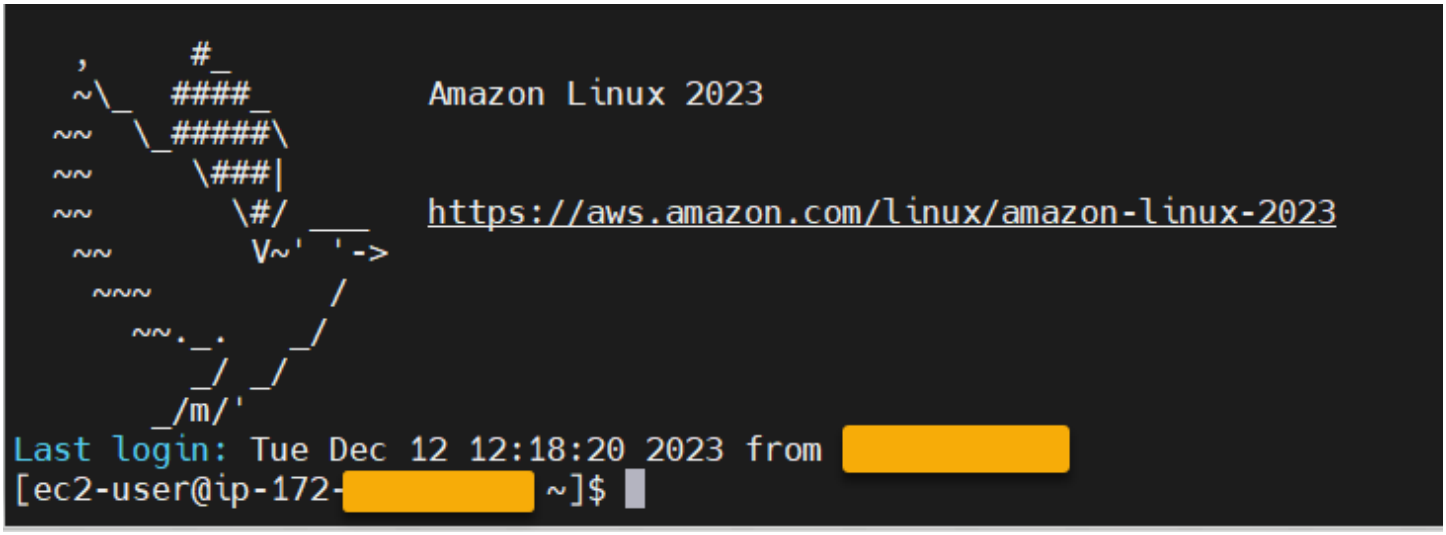
# Starten Sie die AWS Mainframe Modernization Replatform mit der NTT DATA-Instanz

1. Öffnen Sie die AWS Marketplace Konsole unter /marketplace. https://console.aws.amazon.com
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 zu starten. EC2 Mit dieser Aktion gelangen Sie zur EC2 Amazon-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 EC2 Amazon-Schlüsselpaare und Linux-Instances im EC2 Amazon-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 EC2 Amazon-Instance handelt, ist es üblich, TCP-Verkehr zu den Ports 86 und 10086 zuzulassen, um die Konfiguration von Rocket Software (ehemals Micro Focus) zu verwalten.
11. Konfigurieren Sie den Speicher für die EC2 Amazon-Instance.
12. Lesen Sie die Zusammenfassung 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 EC2 Amazon-Konsole unter https://console.aws.amazon.com/ec2/.
15. Wählen Sie im linken Navigationsbereich unter dem Menü Instances die Option Instances aus.

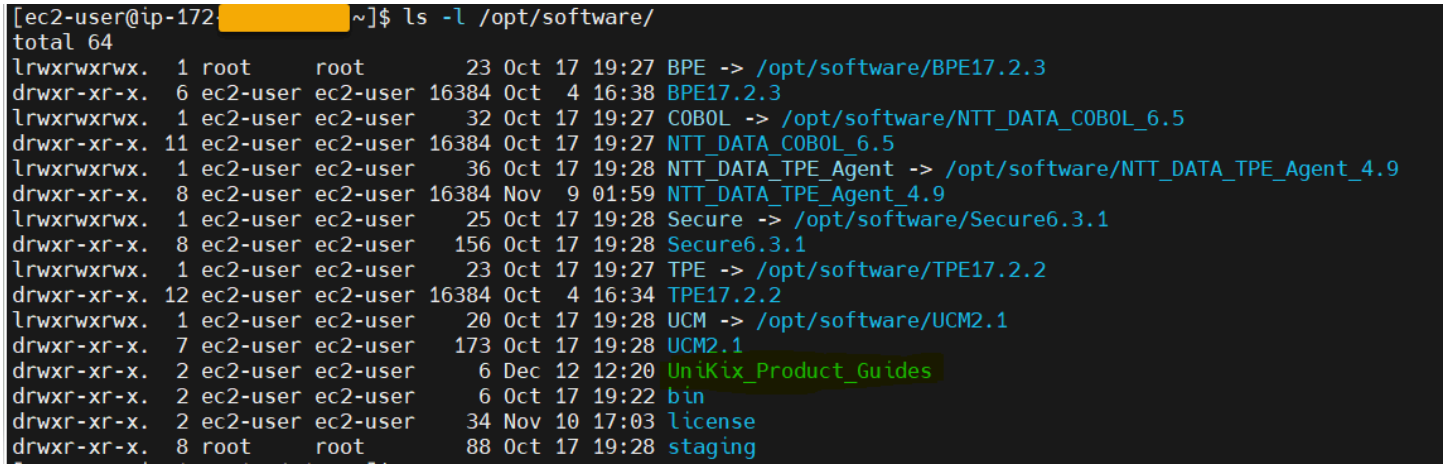
16. Überprüfen Sie im Hauptbereich den Status Ihrer Instance.

# Erste Schritte mit NTT Data

Nachdem Sie die EC2 Amazon-Instance bereitgestellt haben, greifen Sie mit dem Benutzernamen `ec2-user` per SSH auf sie zu. Der Bildschirm wird wie in der folgenden Abbildung aussehen.




Unter dem `/opt/software/` Ordner befindet sich ein Ordner mit dem Namen`UniKix_Product_Guides`, wie in der folgenden Abbildung gezeigt.




Der `UniKix_Product_Guides` Ordner enthält die Dokumentation für die folgenden Komponenten, die auf dieser EC2 Amazon-Instance installiert sind:

- NTT-DATENTYP
- NTT-DATEN, BPE

- NTT DATA COBOL für Unternehmen
- 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 EC2 Amazon-Instance erfolgreich validiert haben, beginnen Sie mit der Nutzung von AWS Mainframe Modernization Replatform mit NTT DATA, indem Sie der NTT Data-Dokumentation folgen.

# Tutorial: Anwendung auf NTT DATA bereitstellen CardDemo

Diese Seite führt Sie durch den step-by-step Prozess zur Bereitstellung der CardDemo Beispielanwendung auf der AWS Mainframe Modernization Replatform mit NTT DATA Unikix Runtime.

Bei der CardDemo Beispielanwendung handelt es sich um eine vereinfachte Mainframe-Anwendung, die entworfen und entwickelt wurde, um Technologien für Anwendungsfälle der Mainframe-Migration und Modernisierung zu testen AWS und zu präsentieren und zu präsentieren.

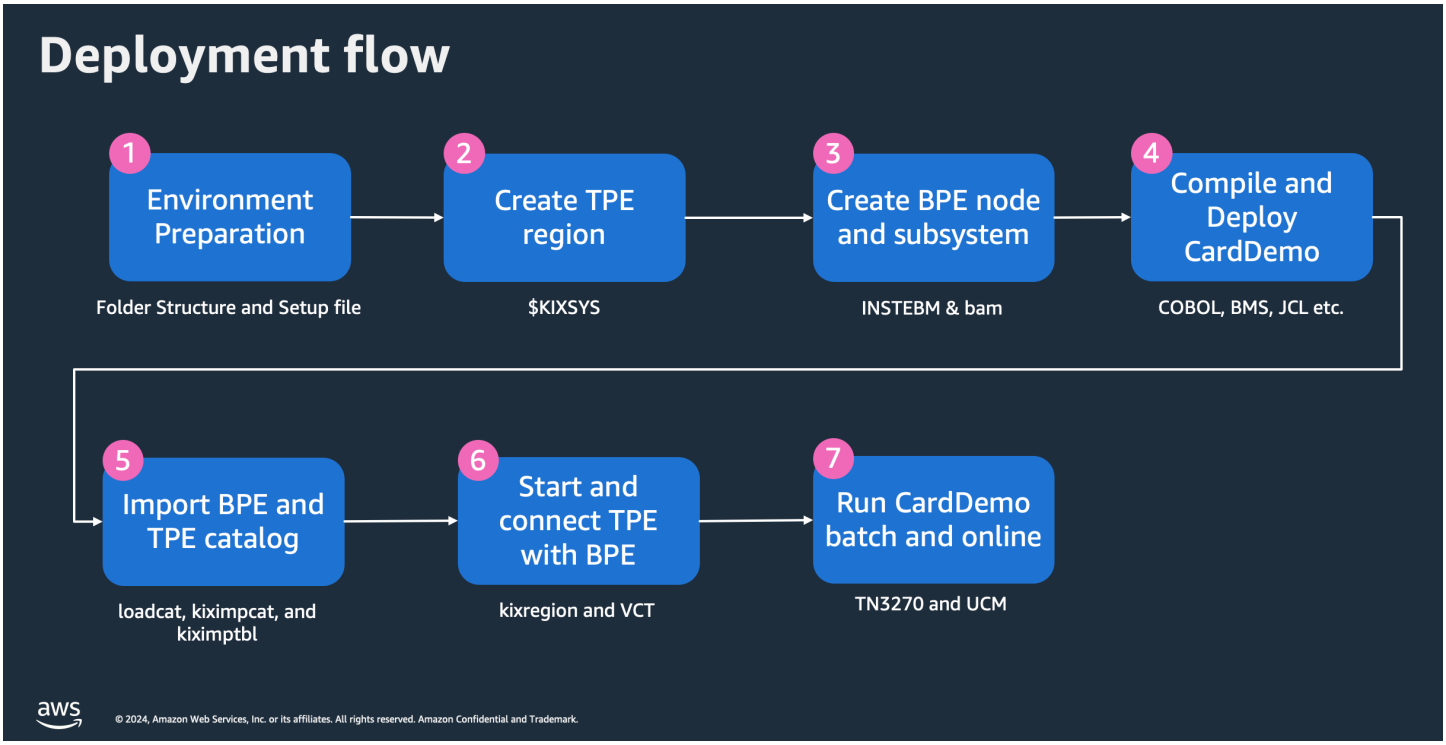
Weitere Informationen zu dieser Anwendung finden Sie unter Repository für. GitHub CardDemo

Themen


- Flussdiagramm für die Bereitstellung
- Voraussetzungen
- Schritt 1: Bereiten Sie die Umgebung vor
- Schritt 2: Erstellen Sie eine TPE-Region
- Schritt 3: Erstellen Sie den BPE-Knoten und das Subsystem
- Schritt 4: Anwendung kompilieren und bereitstellen CardDemo
- Schritt 5: BPE- und TPE-Katalog importieren
- Schritt 6: Starten und verbinden Sie TPE mit BPE
- Schritt 7: Führen Sie die Anwendung aus CardDemo
- Fehlerbehebung

# Flussdiagramm für die Bereitstellung

Das folgende Diagramm zeigt jeden Schritt im Workflow für die Bereitstellung einer Anwendung auf der NTT DATA Unikix-Laufzeit.




## Voraussetzungen

- Folgen Sie den Anweisungen unter Umgestaltung der Plattform mit NTT DATA Verwenden des NTT DATA UniKix Marketplace AMI.
- Ändern Sie die Option für Instance-Metadaten IMDSv2auf Optional, wie unter Restore the use of IMDSv1 im  EC2 Amazon-Benutzerhandbuch beschrieben.
- Laden Sie die CardDemo Laufzeitkomponenten für NTT DATA UniKix von der

  GitHub Repository.
- Melden Sie sich bei der UniKix EC2 Laufzeitinstanz an als`ec2-user`.
- Extrahieren Sie die heruntergeladenen CardDemo Runtime-Komponenten über diesen Link: UniKix_ CardDemo _runtime_v1.zip.
  - Das entpackte Verzeichnis sollte zwei `migrated_app` Verzeichnisse enthalten`bin`.
  - Verschieben Sie `bin` sowohl als auch `migrated_app` Verzeichnisse in Ihr `$HOME` Verzeichnis. Der Pfad würde so aussehen`/home/ec2-user`.

- Sie sollten die folgenden Verzeichnisse in Ihrem haben$HOME:
  - /home/ec2-user/bin
  - /home/ec2-user/migrated_app
- Verschieben Sie alle Dateien im Verzeichnis $HOME/bin mit dem folgenden Befehl:
- • chmod +x $HOME/bin/*

## Schritt 1: Bereiten Sie die Umgebung vor

Nachdem Sie die Voraussetzungen erfüllt haben, besteht der erste Schritt darin, die Umgebung vorzubereiten, in der Sie die CardDemo Anwendung bereitstellen möchten.

1. Melden Sie sich bei der UniKix EC2 Runtime-Instanz an alsec2-user.
2. Beachten Sie die Liste der im AMI vorkonfigurierten UniKix Softwarepakete wie TPE, BPE und COBOL sowie andere Software aus dem NTT UniKix DATA-Produktverzeichnis, indem Sie in Ihrer Instanz den folgenden Befehl verwenden: EC2

   ```
   ls -l /opt/software/
   ```

3. Untersuchen Sie die migrierte Anwendung. CardDemo Sie werden den gesamten Quellcode sehen, einschließlich BMS-Maps, COBOL-Programmen, COBOL-Copybooks und.
   JCLs Auf diese Weise finden Sie auch den Export von BPE- und TPE-Katalogen, CICS-Ressourcendefinitionen und den migrierten Daten wie sequentiellen Dateien und VSAM-Dateien:

   ```
   ls  $HOME/migrated_app/*/*
   ```

4. Erstellen Sie eine Ordnerstruktur, indem Sie das Skript mit dem folgenden Befehl ausführen: create_project

   ```
   sh $HOME/bin/create_project
   ```

5. Aktivieren Sie die CardDemo Umgebung, indem Sie die carddemo.env Setup-Datei wie folgt beziehen:

   ```
   source $HOME/bin/carddemo.env
   ```

# Schritt 2: Erstellen Sie eine TPE-Region

Nachdem Sie die Umgebung aktiviert haben, in der Sie die Anwendung bereitstellen möchten, müssen Sie eine TPE-Region erstellen.

1. Erstellen Sie eine TPE-Region mit dem `kixregion createRegion` Befehl, der Eingaben wie `$KIXSYS$JAVA_HOME`, und erfordert. `$KIXLICDIR` Diese Umgebungsvariablen sind bereits in der `carddemo.env` Setup-Datei eingerichtet.

```
kixregion createRegion $KIXSYS $JAVA_HOME $KIXLICDIR
```

2. Konfigurieren Sie die TPE-Region mit dem `kixregion setAttr` Befehl.

```
kixregion setAttr $KIXSYS server.tx.languages.cobol.enabled true
kixregion setAttr $KIXSYS server.tx.languages.cobol.flavor vcobol
kixregion setAttr $KIXSYS server.tx.languages.cobol.home $VCOBOL
kixregion setAttr $KIXSYS maps.location $PROJECT_ROOT/maps
kixregion setAttr $KIXSYS programs.location $PROJECT_ROOT/loadlib
kixregion setAttr $KIXSYS environment.KIXDATA $KIXDATA
kixregion setAttr $KIXSYS td.jobq.submission.node $EBMHOME
kixregion setAttr $KIXSYS td.jobq.submission.subsys $EBMSYS
```

3. Generieren Sie die für diese TPE-Region spezifische Benutzerumgebungsdatei, indem Sie den `kixregion createScript` Befehl ausführen. Dieser Befehl erstellt oder aktualisiert `$KIXSYS/bin/userenv` basierend auf der TPE-Regionskonfiguration.

```
kixregion createScript $KIXSYS
```

4. Aktivieren Sie die TPE-Region, indem Sie die Benutzerumgebungsdatei () `$KIXSYS/bin/userenv` abrufen.

```
source $KIXSYS/bin/userenv
```

5. Erstellen Sie die TPE-Region, indem Sie den `kixinstall2` Befehl ausführen.

```
kixinstall2
```

# Schritt 3: Erstellen Sie den BPE-Knoten und das Subsystem

Nachdem Sie eine TPE-Region erstellt haben, müssen Sie den BPE-Knoten und das BPE-Subsystem erstellen, indem Sie die folgenden Schritte ausführen.

1. Ändern Sie den Besitz und die Berechtigungen von. `INSTEBM`

```
sudo chown root $INSTEBM
sudo chmod 4755 $INSTEBM
```

2. Erstellen Sie mit dem `INSTEBM` Befehl einen BPE-Knoten. Das BPE-Knotenverzeichnis wird als Eingabeparameter bereitgestellt.

```
$INSTEBM $EBMHOME
```

3. Aktivieren Sie die Batch-Umgebung, indem Sie die `batchenv` Datei vom neu erstellten BPE-Knoten beziehen.

```
source $EBMHOME/batchenv
```

4. Erstellen Sie das BPE-Subsystem innerhalb dieses Knotens mit dem Batch Administration Manager (bam). Der `bam` Befehl öffnet die Batch Administration Manager-Oberfläche.

```
bam
```

   a. Starten Sie den BPE-Knoten über die BAM-Schnittstelle. Wählen Sie im Hauptmenü Option 2, Systemumgebungen.

```
                Batch Administration Manager                2024/09/17 21:25:56




             1   Software License Management
             2   System Environments
             3   Applications & Subsystems
             4   Security & Users
             5   Classes & Activities
             6   Problem Determination



             H - Help
             Q - Quit

--------------------------------------------------------------------------------
Type an option and press Return: 2
```

b. Wählen Sie Option 2, Start/ (Stop) Batch Node, um den BPE-Knoten zu starten.

```
                System Environments                         2024/09/17 21:27:03

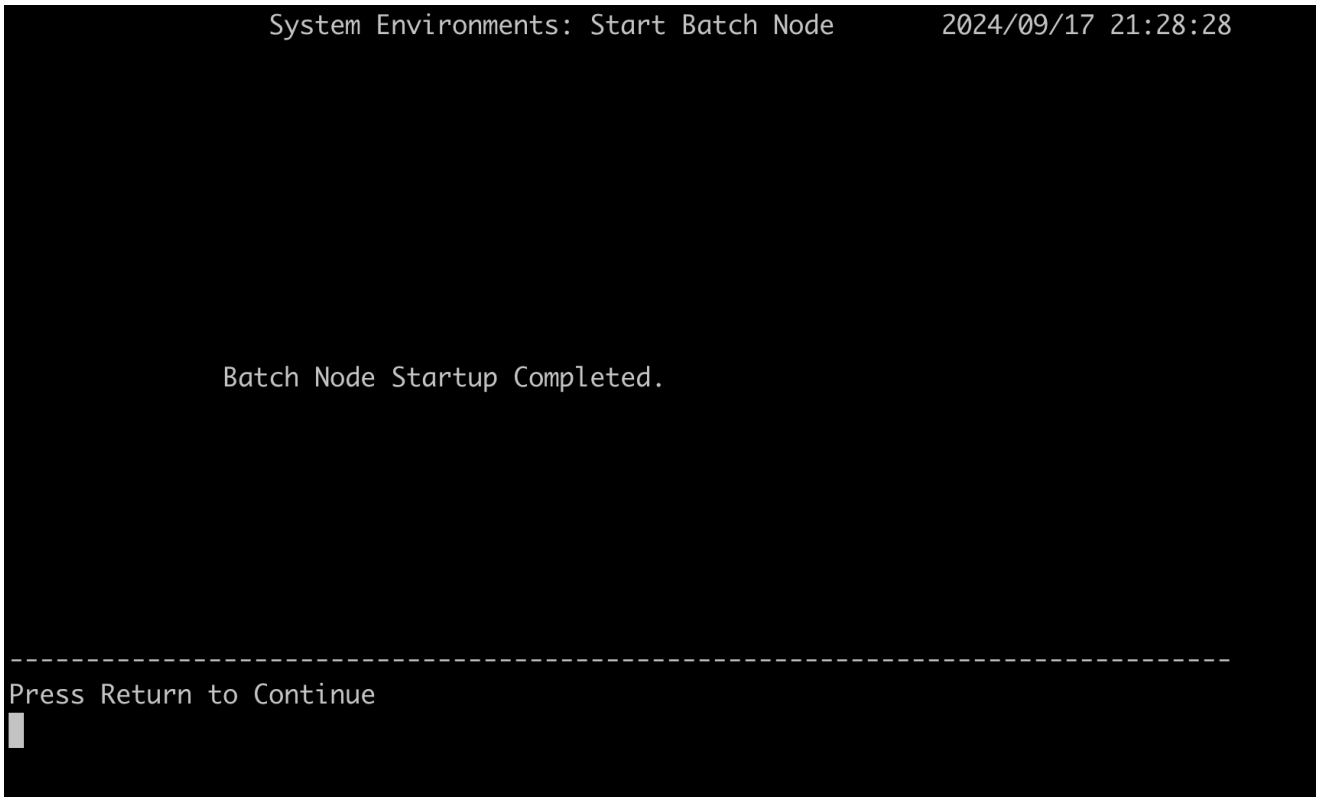



             1   Report System Status
             2   Start/(Stop) Batch Node
             3   Assign the Console
             4   Change the Date
             5   Redirect Job Output
             6   Inter-Node Communications
             7   Job Accounting
             8   Assign the Initial Job Number
             9   Enable/Disable Duplicate Name Execution Delay : Enabled

             H - Help
             R - Return to Main Menu

--------------------------------------------------------------------------------
Type an option and press Return: 2
```

c. Drücken Sie nach dem Start zweimal die Eingabetaste, um zum BAM-Hauptmenü zurückzukehren.




d. Um das BPE-Subsystem zu erstellen, wählen Sie Option 3, Anwendungen und Subsysteme.

```
                Batch Administration Manager                2024/09/17 21:29:03




             1   Software License Management
             2   System Environments
             3   Applications & Subsystems
             4   Security & Users
             5   Classes & Activities
             6   Problem Determination



             H - Help
             Q - Quit

--------------------------------------------------------------------------------
Type an option and press Return: 3
```

e. Wählen Sie dann Option 3, Subsystem erstellen.

```
                Applications & Subsystems                   2024/09/17 21:29:57




             1   List All Subsystems
             2   Query a Subsystem
             3   Create a Subsystem
             4   Update a Subsystem
             5   Delete a Subsystem
             6   Change/Show Default Subsystem
             7   Import a Subsystem (from another batch node)
             8   Create a BPESUB Project
             9   Export Subsystem bldsub Config File

             H - Help
             R - Return to Main Menu

--------------------------------------------------------------------------------
Type an option and press Return: 3
```

f. Geben Sie den Namen des Subsystems als ein. `sys1`

```
                Applications & Subsystems: Create          2024/09/17 21:30:22









         No Subsystems are currently defined.









             R - Return to Previous Screen

--------------------------------------------------------------------------------
Enter the Subsystem's name you want to create: sys1
```

g. Wählen Sie Option 3, Datenverwaltung.

```
                Applications & Subsystems: Create          2024/09/17 21:30:53



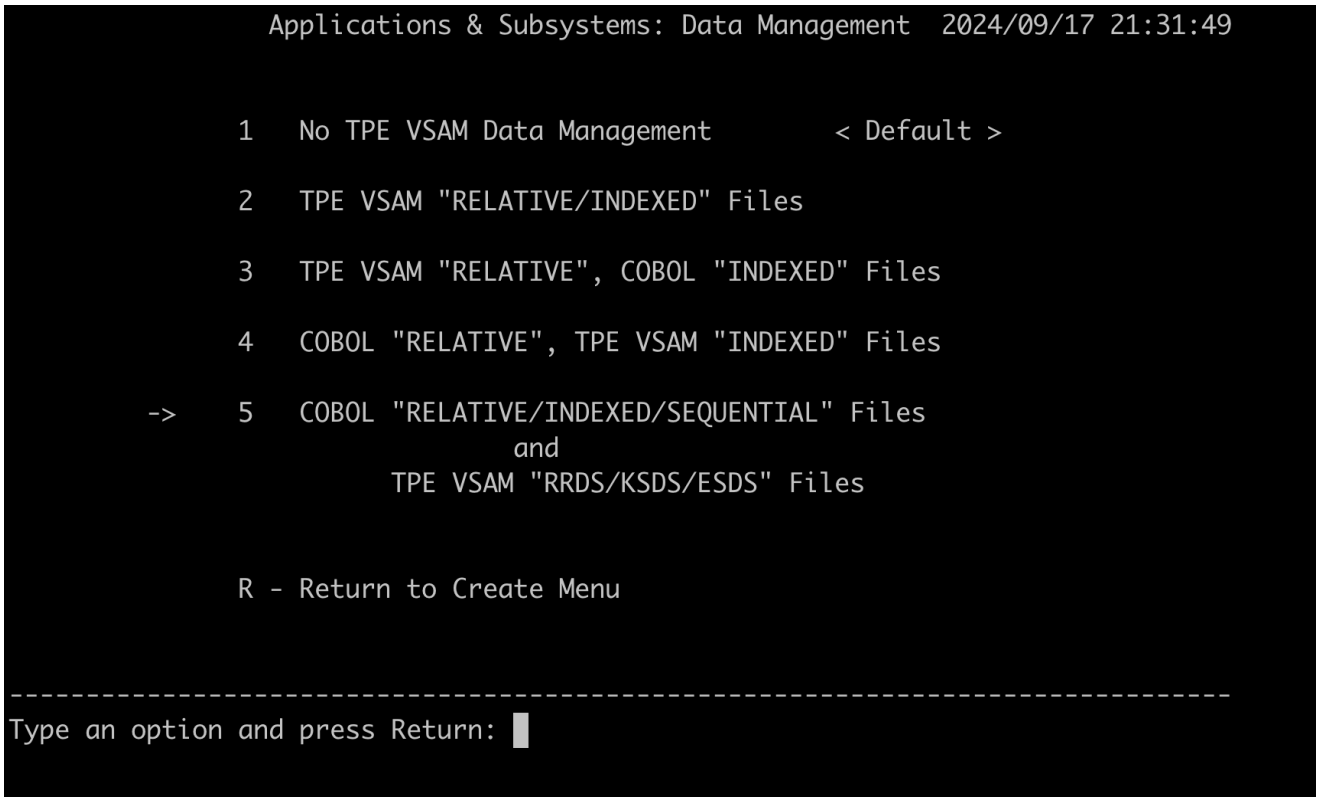
             D   Display Current sys1 Subsystem's Configuration
             S   Set to Default Subsystem Configuration

             1   Application Languages
             2   Database Management System (DBMS)
             3   Data Management
             4   Optional Packages
             5   Date/Time Management
             6   User-Specific Objects
             7   Configuration Options

             C - Create sys1 Subsystem
             H - Help
             R - Return to Main Menu

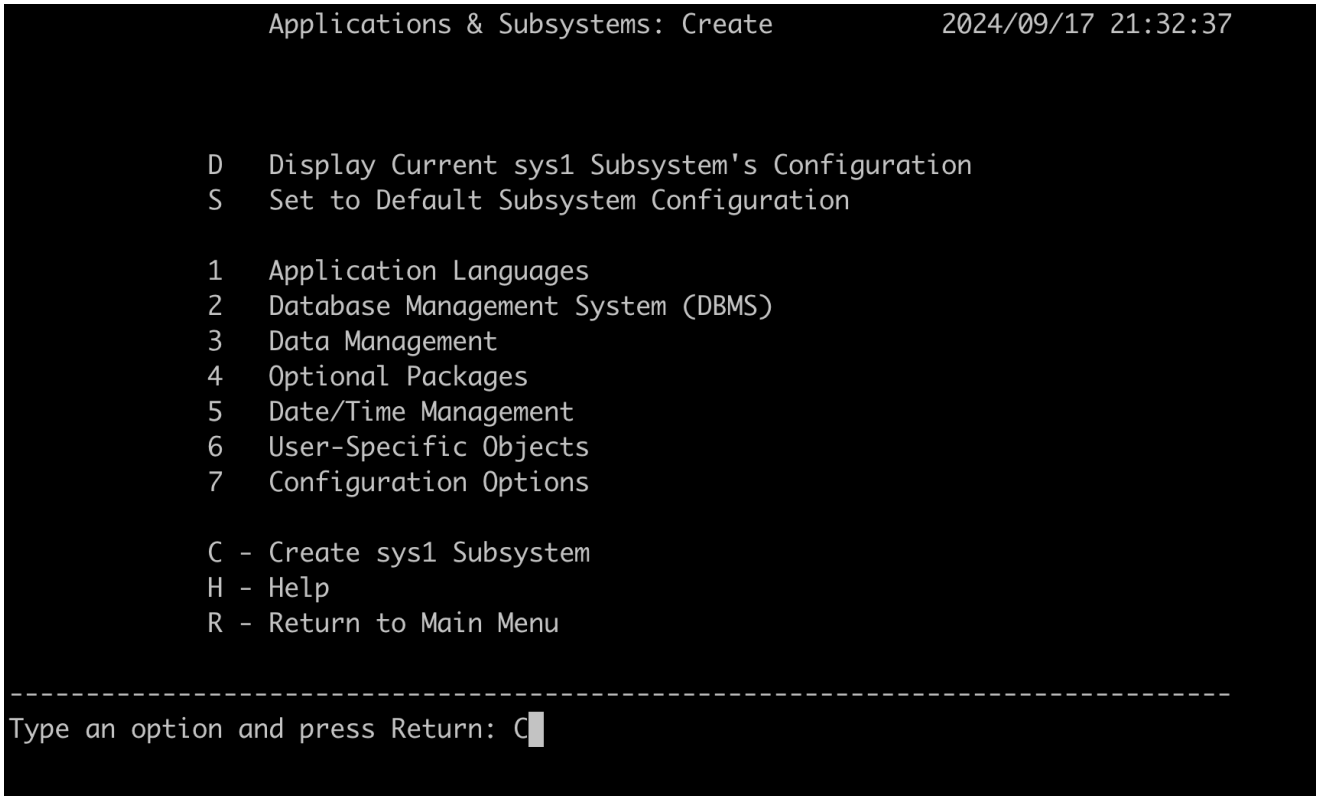
--------------------------------------------------------------------------------
Type an option and press Return: 3
```

h. Wählen Sie Option 5, da die CardDemo Anwendung sowohl sequentielle als auch VSAM-Dateien umfasst.



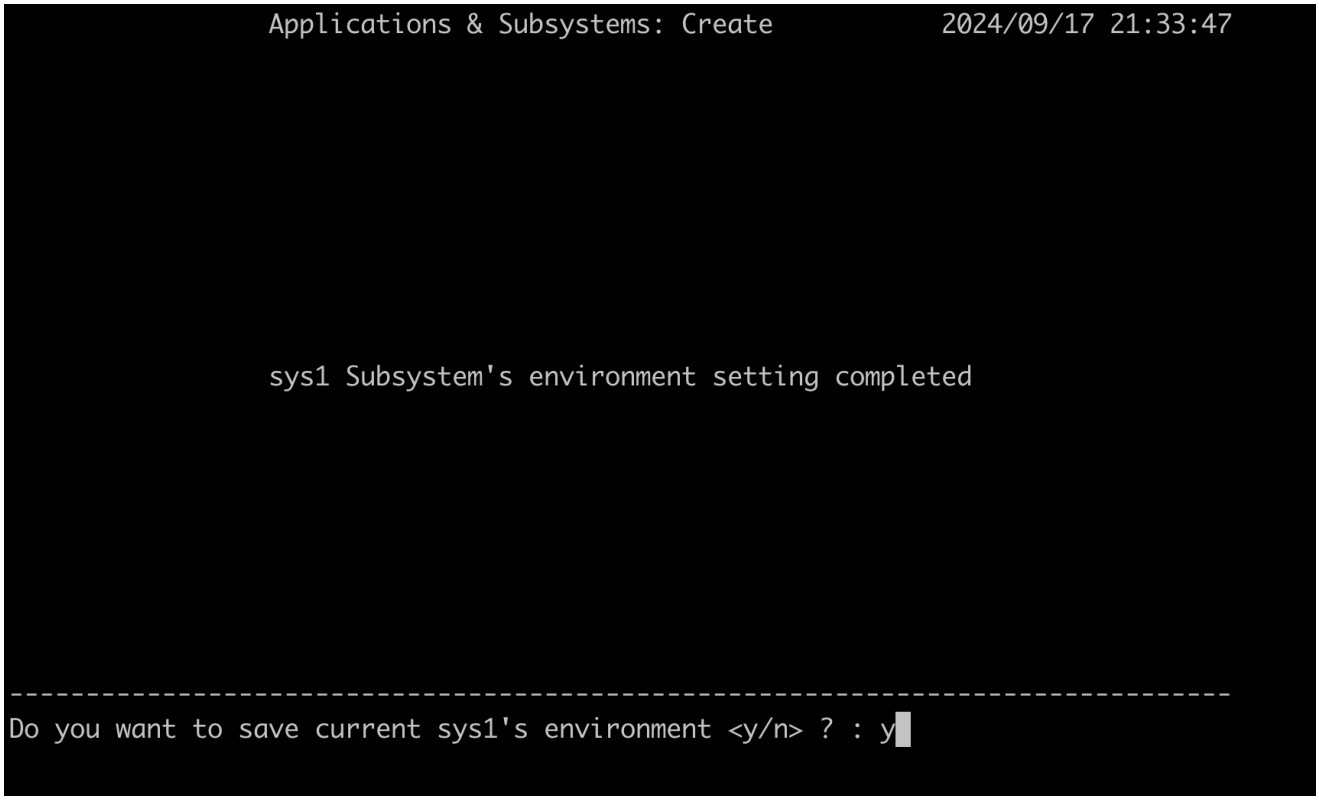

i. (Optional). Drücken Sie „R“, um zur Seite „Menü erstellen“ zurückzukehren und die verschiedenen verfügbaren Konfigurationsoptionen zu überprüfen.

j. Geben Sie auf der Seite „Erstellen“ „C“ ein, um das Subsystem `sys1` zu erstellen.

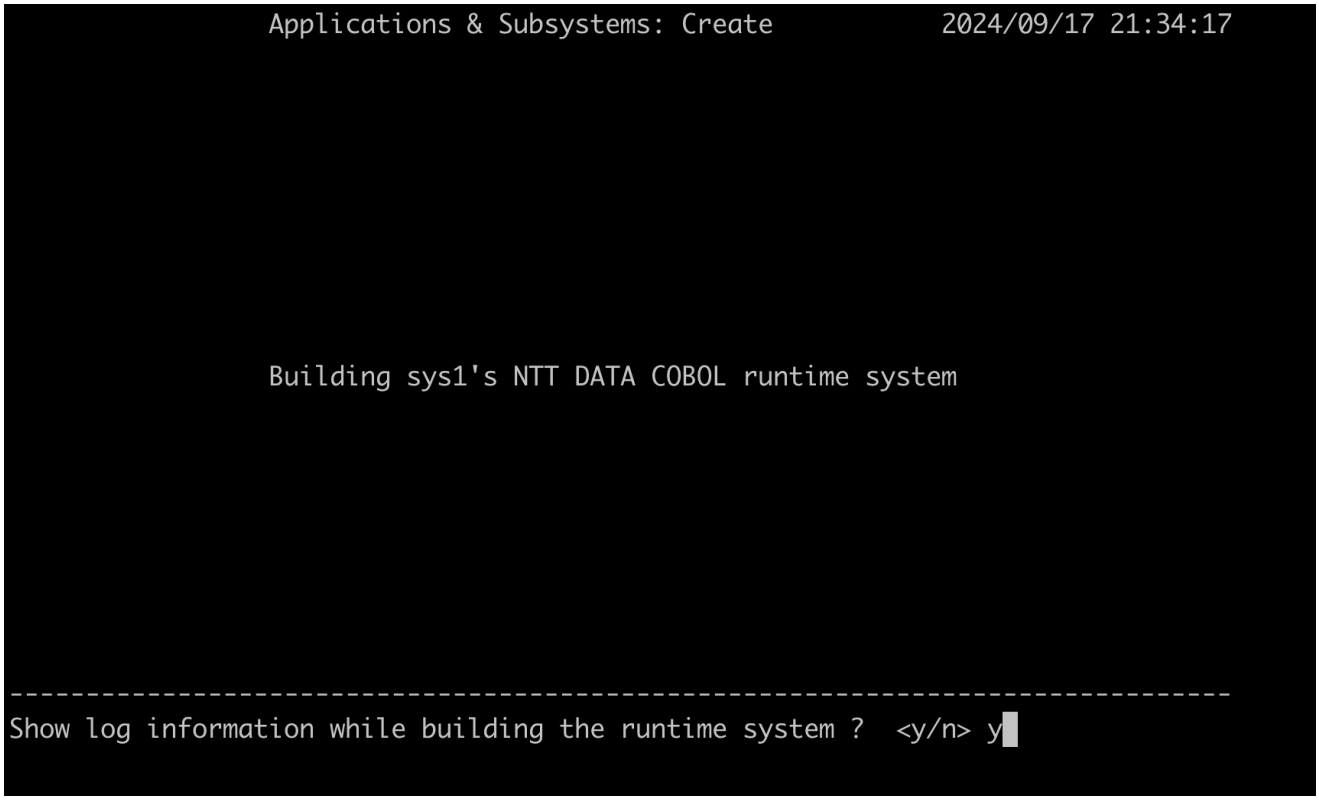



k. Überprüfen Sie die Einstellungen und geben Sie „C“ ein, um mit den restlichen Umgebungseinstellungen fortzufahren. Diese Umgebungseinstellungen sind vorbelegt, da die erforderlichen Umgebungsvariablen in der `carddemo.env` Setup-Datei definiert sind und die empfohlene Ordnerstruktur vorhanden ist.

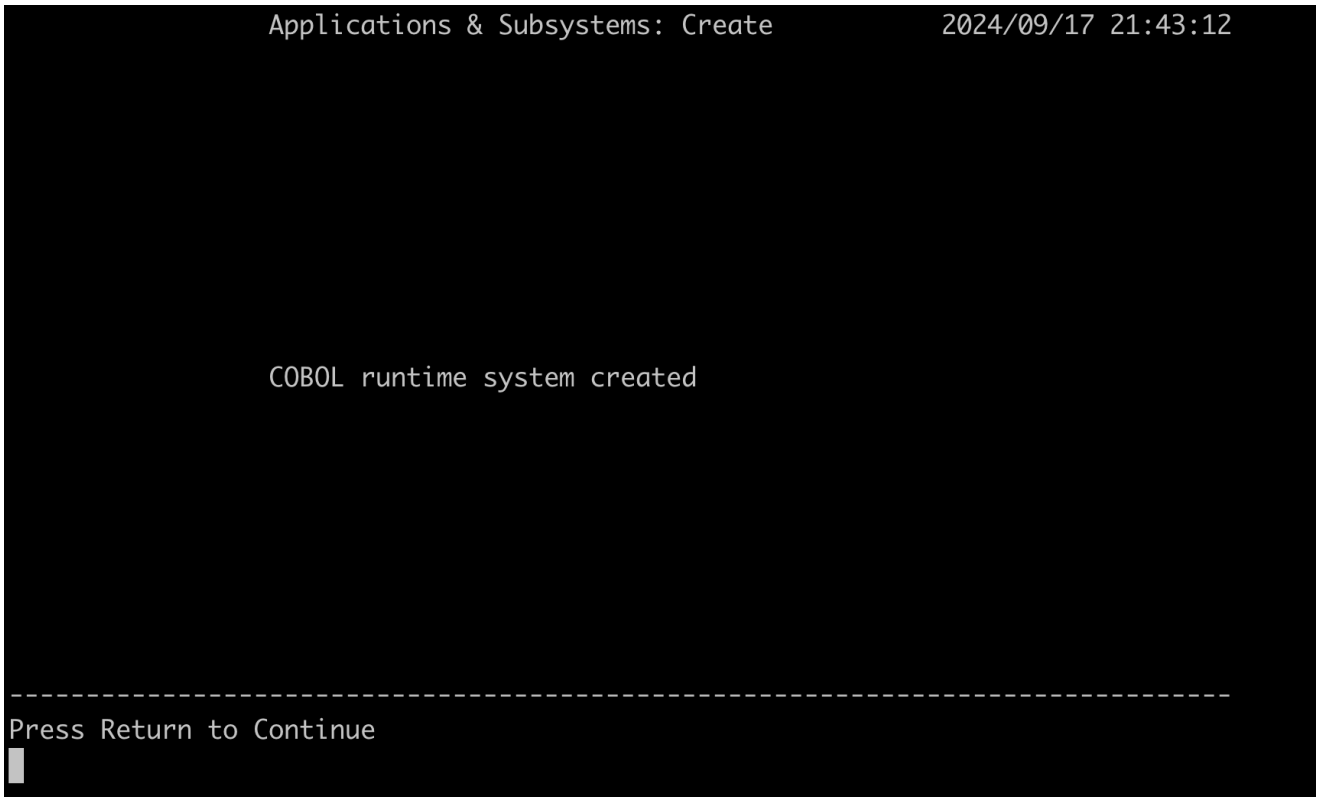
l. Geben Sie „y“ ein, um die aktuellen Umgebungseinstellungen zu bestätigen und zu speichern.

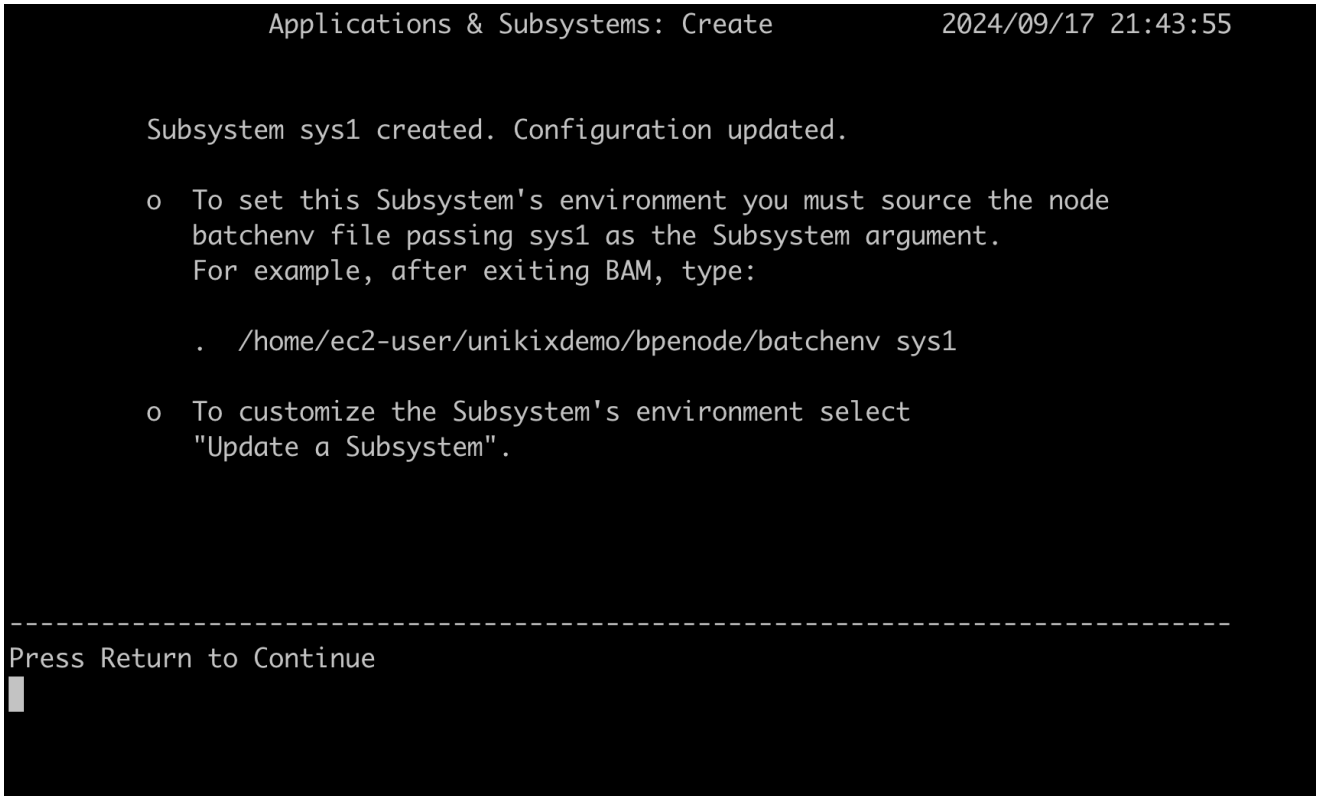


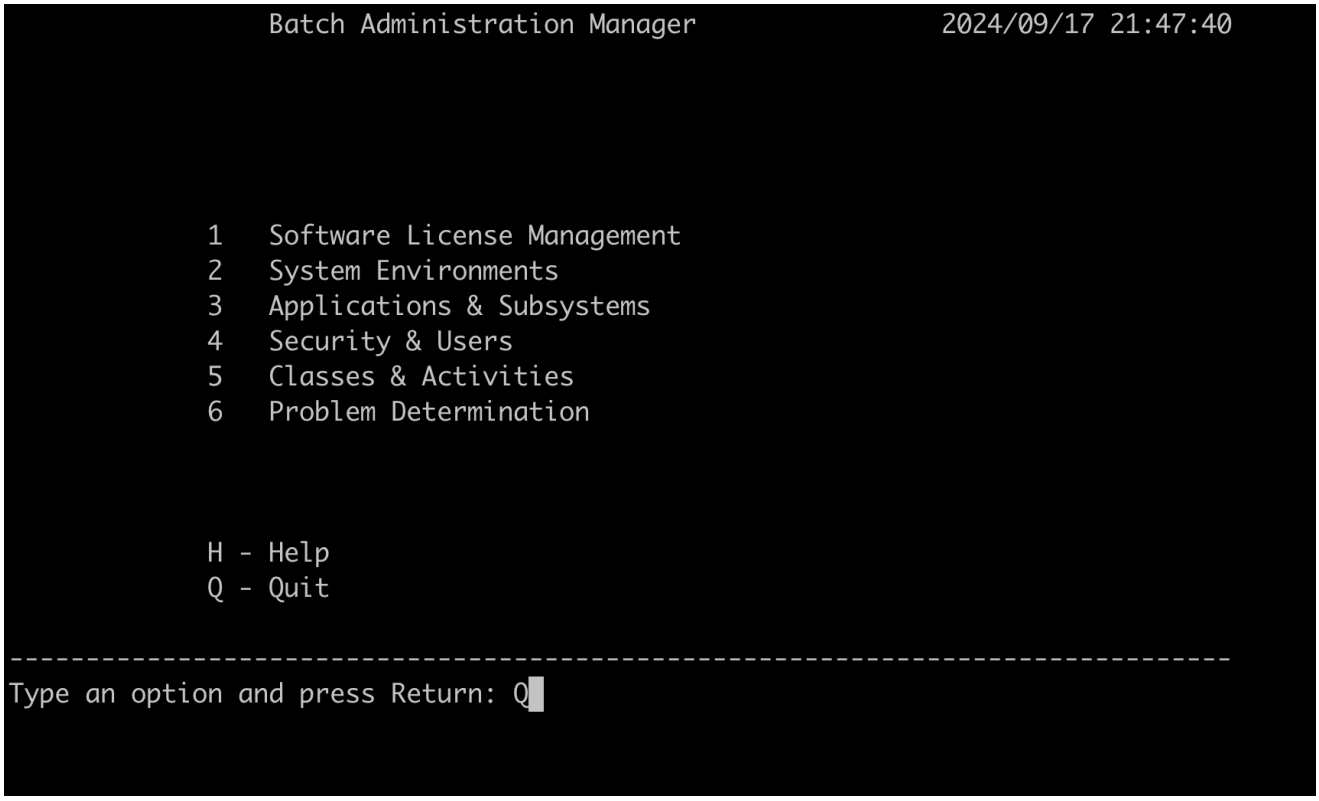

m. Geben Sie „y“ ein, um das Protokoll beim Aufbau des Subsystems anzuzeigen.



n. Drücken Sie die Eingabetaste, bis Sie zum Hauptmenü zurückkehren, und verlassen Sie die BAM-Oberfläche, indem Sie die Option Quit wählen.






5. Aktivieren Sie das BPE-Subsystem, indem Sie das `batchenv` mit dem Namen des Subsystems beziehen. `sys1`

```
source $EBMHOME/batchenv sys1
```

## Schritt 4: Anwendung kompilieren und bereitstellen CardDemo

In diesem Schritt kompilieren Sie die COBOL-Programme und stellen Anwendungsartefakte wie JCL, Prozeduren, Datendateien und CICS-Ressourcendefinitionen bereit.

1. Aktivieren Sie die CardDemo Umgebung erneut, indem Sie die Setup-Datei abrufen`carddemo.env`.

```
source $HOME/bin/carddemo.env
```

2. Navigieren Sie zum COBOL-Quellverzeichnis.

```
cd $MIGAPP_DIR/cbl
```

3. Kompilieren Sie das Cobol-Programm `CBACT01C.cbl` mithilfe eines Skripts`compile`.

```
compile CBACT01C.cbl
```

4. Kompilieren Sie alle Cobol-Programme mithilfe eines Skripts`compile.all`.

```
compile.all
```

5. Navigieren Sie zum Quellverzeichnis von BMS Maps.

```
cd $MIGAPP_DIR/bms
```

6. Kompilieren Sie die BMS-Map `COACTUP.bms` mithilfe eines Skripts`compbms`.

```
compbms COACTUP.bms
```

7. Kompilieren Sie alle BMS-Karten mithilfe eines Skripts`compbms.all`.

```
compbms.all
```

8. Überprüfen Sie die kompilierten Binärdateien für COBOL- und BMS-Karten.

```
ls $PROJECT_ROOT/loadlib
ls $PROJECT_ROOT/maps
```

9. Stellen Sie mithilfe des Skripts andere Anwendungsartefakte wie JCL, Prozeduren, Datendateien und CICS-Ressourcendefinitionen bereit. `deploy_app`

```
deploy_app
```

10. Navigieren Sie zum JCL-Verzeichnis des Projekts.

```
cd $PROJECT_ROOT/jcl
```

11. Translate Sie JCL ACCTFILE in BPE JCL Macro. Verwenden Sie den `mvstrans` Befehl und verwenden Sie dabei die Option „-v“ für die JCL-Überprüfung und die Option „-f“, um das Makro zu erstellen.

```
mvstrans ACCTFILE -v
mvstrans ACCTFILE -f
```

12. Translate Sie die JCL-Prozedur REPROC in die BPE JCL-Prozedur Macro. Verwenden Sie den `mvstrans` Befehl mit der Option „-p“ zusammen mit der Option „-v“ zur Überprüfung und der Option „-f“, um das Makro zu erstellen.

```
mvstrans REPROC -v -p
mvstrans REPROC -f -p
```

13. Translate Sie alle JCLs und JCL-Prozeduren.

```
for file in "./jmvs/*"; do mvstrans $file -f; done > jmvs.out
for file in "./mvsp/*"; do mvstrans $file -p -f; done > mvsp.out
```

## Schritt 5: BPE- und TPE-Katalog importieren

In diesem Schritt importieren Sie den BPE- und TPE-Katalog mit unterschiedlichen Befehlen.

1. Importieren Sie den BPE-Katalog mithilfe eines Befehls. `loadcat`

```
loadcat $MIGAPP_DIR/catlg/bpe/BPECAT*
```

2. Navigieren Sie zum Verzeichnis $KIXSYS.

```
cd $KIXSYS
```

3. Importieren Sie den TPE-Katalog mit dem Befehl. `kiximpcat`

```
kiximpcat -c CATALOG -l CATALOG.lst
```

4. Importieren Sie CICS-Ressourcendefinitionen mit dem Befehl kiximptbl.

```
kiximptbl
```

## Schritt 6: Starten und verbinden Sie TPE mit BPE

In diesem Schritt müssen Sie die zuvor erstellte TPE-Region zusammen mit dem BPE-Manager starten und diese verbinden, um die Beispielanwendung ausführen zu können. CardDemo

1. Führen Sie den `kixverify` Befehl für alle VSAM-Dateien aus, um sicherzustellen, dass sie zurückgesetzt und alle zuvor geöffneten Dateien geschlossen wurden.

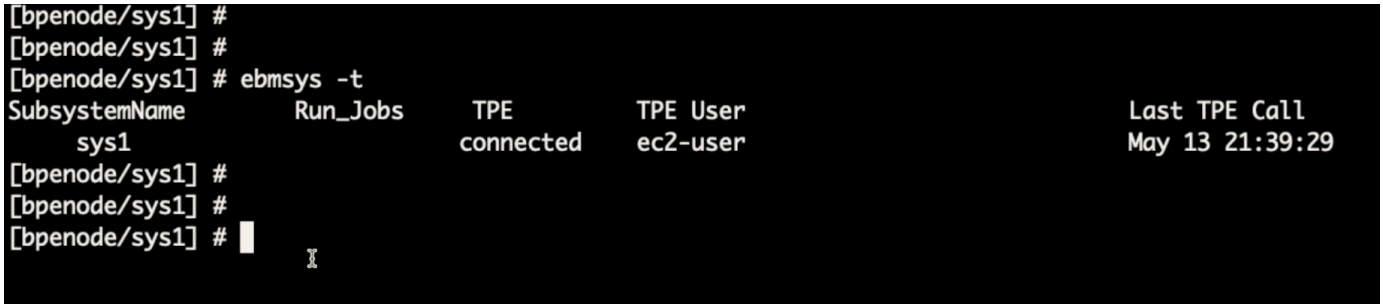
```
kixverify -r ALL
```

2. Starten Sie die TPE-Region.

```
kixregion start $KIXSYS
```

3. Stellen Sie sicher, dass sowohl BPE als auch TPE angeschlossen sind. Dies ist von entscheidender Bedeutung, da die VSAM-Dateien Eigentum von TPE sind und für jeden Batch-Vorgang, der auf das VSAM zugreift, eine Verbindung zu TPE erforderlich ist.

```
ebmsys -t
```




## Schritt 7: Führen Sie die Anwendung aus CardDemo

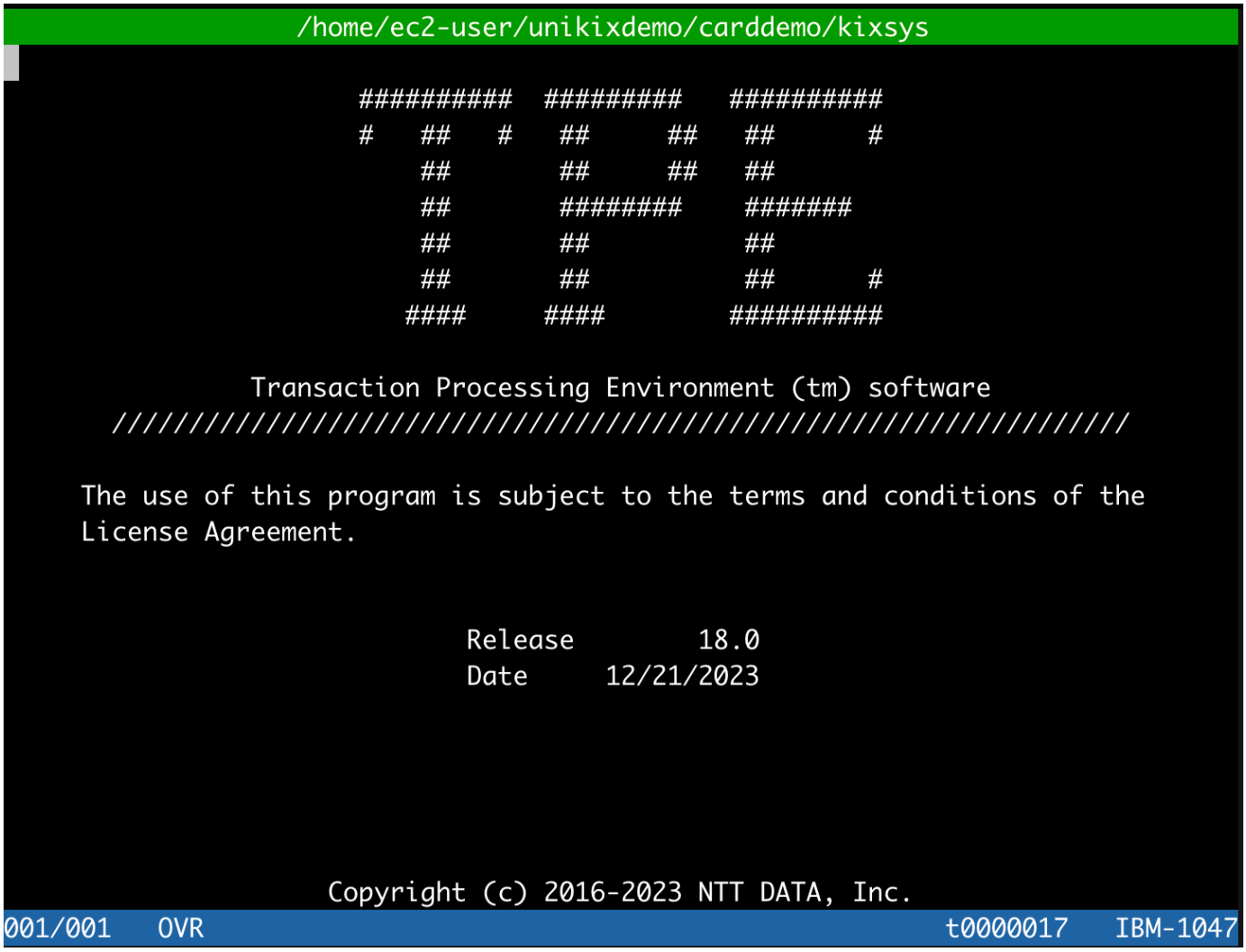
In diesem Schritt führen Sie die CardDemo Anwendung im TN327 0-Terminal-Emulator aus.

Das UniKix Runtime-AMI enthält einen TN327 0-Terminal-Emulator, den Sie direkt von der UniKix EC2 Instance aus starten können.

Stellen Sie mithilfe des TN327 0-Terminalemulators eine Connect zu TPE her

- Starten Sie das TN327 0-Terminal mit dem `kixterm` Befehl.

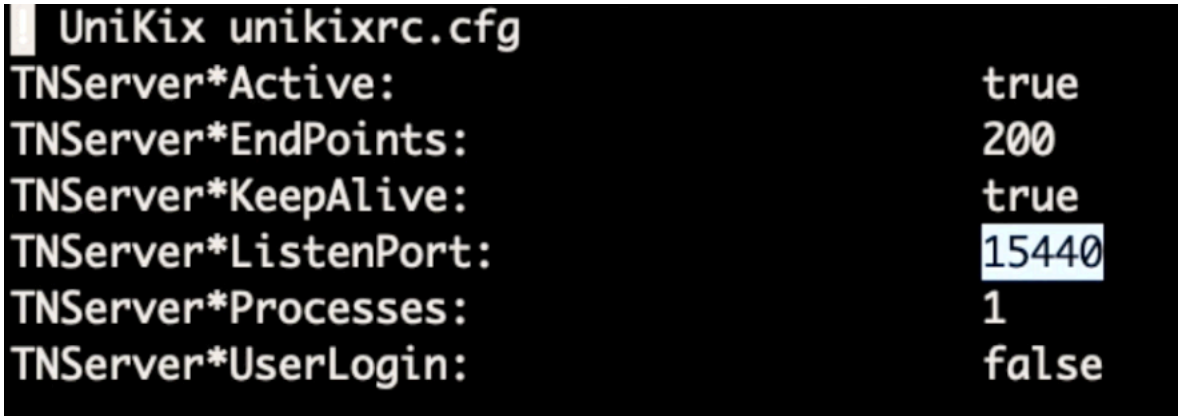
```
kixterm
```




(Optional). Wenn Sie Ihren eigenen Terminalemulator verwenden möchten:

1. Rufen Sie die IP-Adresse der UniKix Runtime-Instance von der EC2 Amazon-Konsole ab.
2. Rufen Sie die Portnummer für die Verbindung mit der TPE-Region mithilfe des TN327 0-Terminal-Emulators ab. Sie finden dies in der TNServer ListenPort Datei unikixrc.cfg.

```
cat $KIXSYS/unikixrc.cfg
```

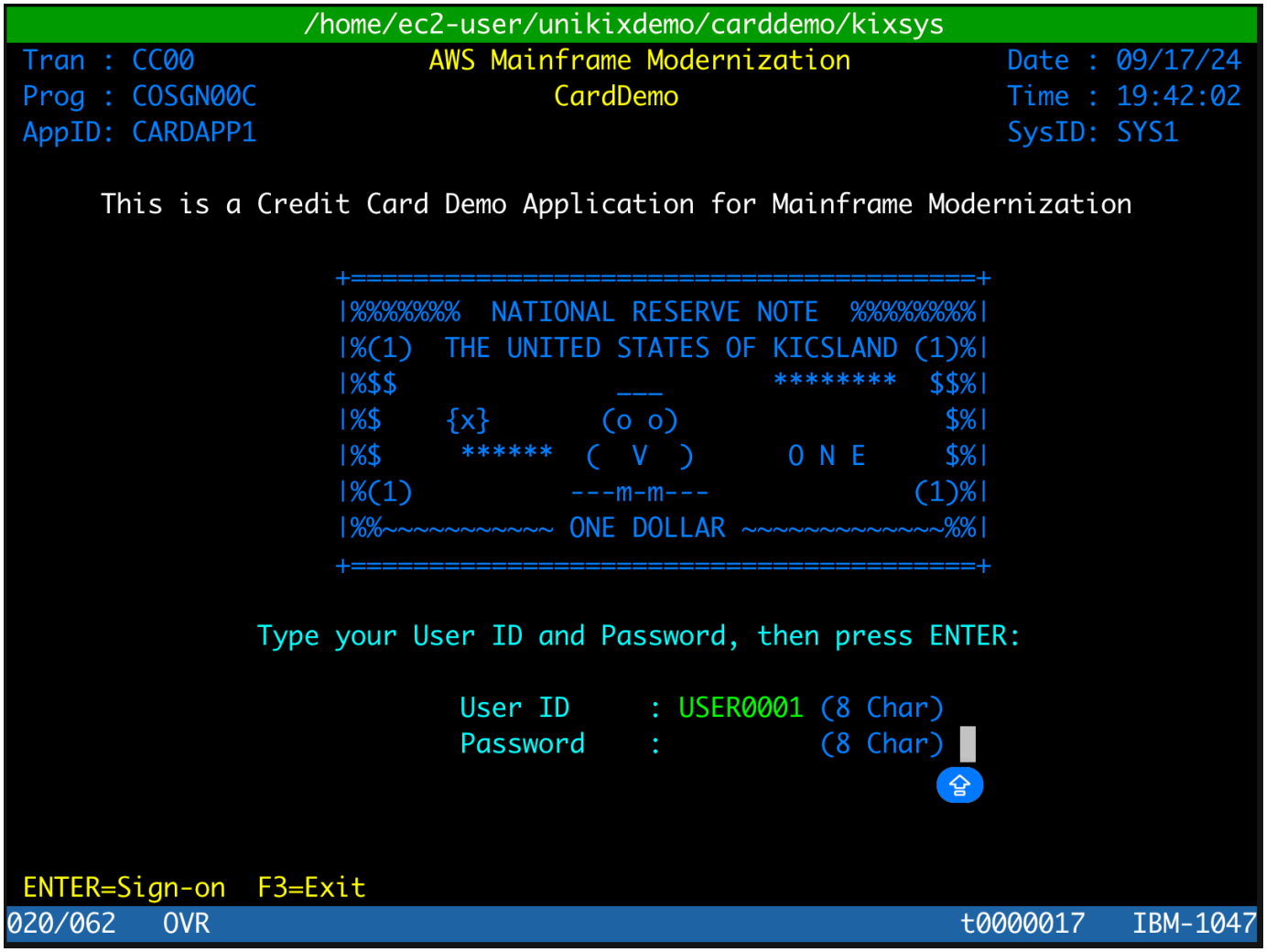



3. Konfigurieren Sie Ihren TN327 0-Terminal-Emulator so, dass er die IP-Adresse der UniKix Laufzeitinstanz und die Portnummer 15440 verwendet.

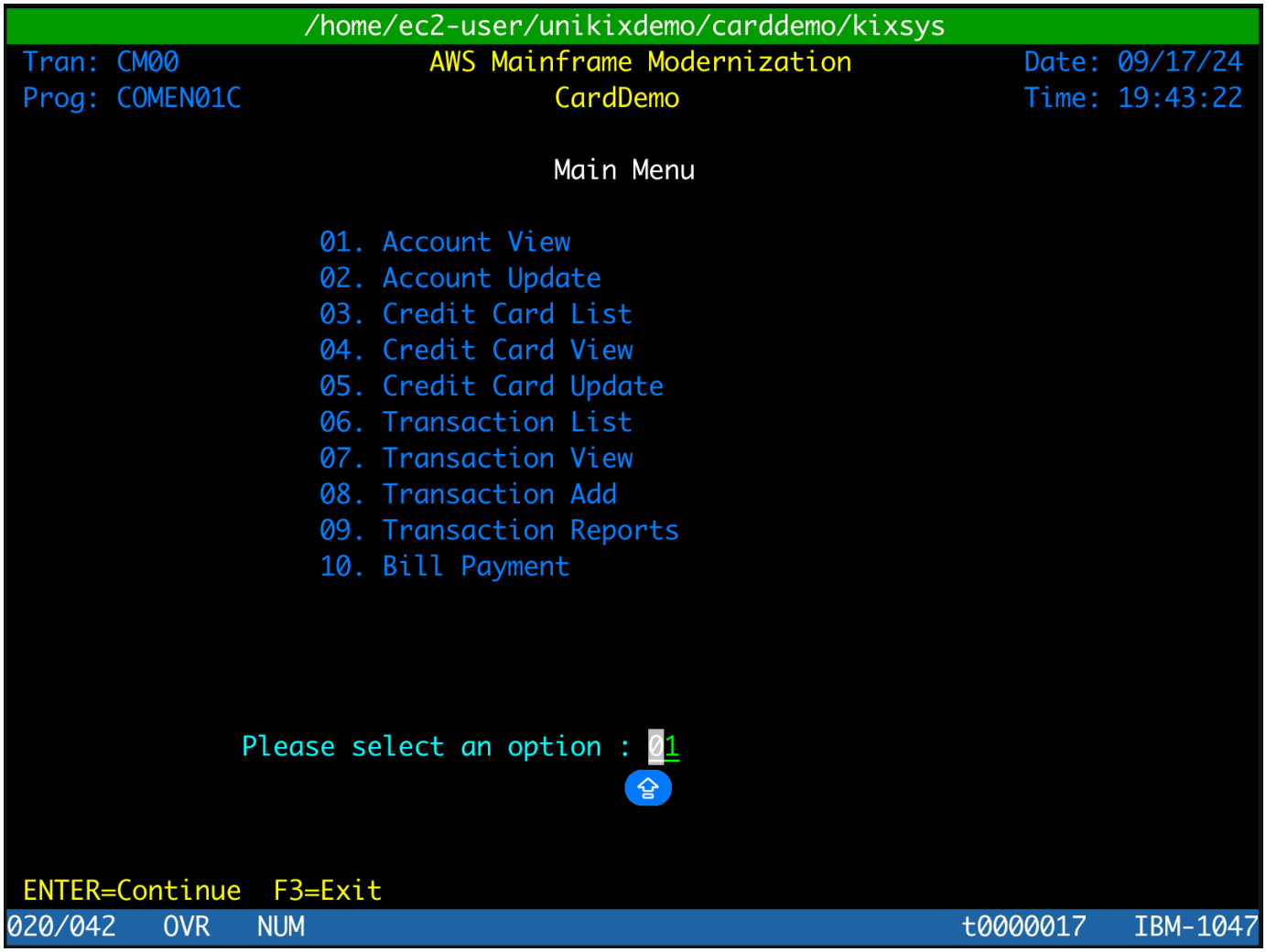
Online-Transaktionen

In diesem Abschnitt wird davon ausgegangen, dass Sie mit dem `kixterm` Befehl eine Verbindung zum TN327 0-Terminal-Emulator hergestellt haben.

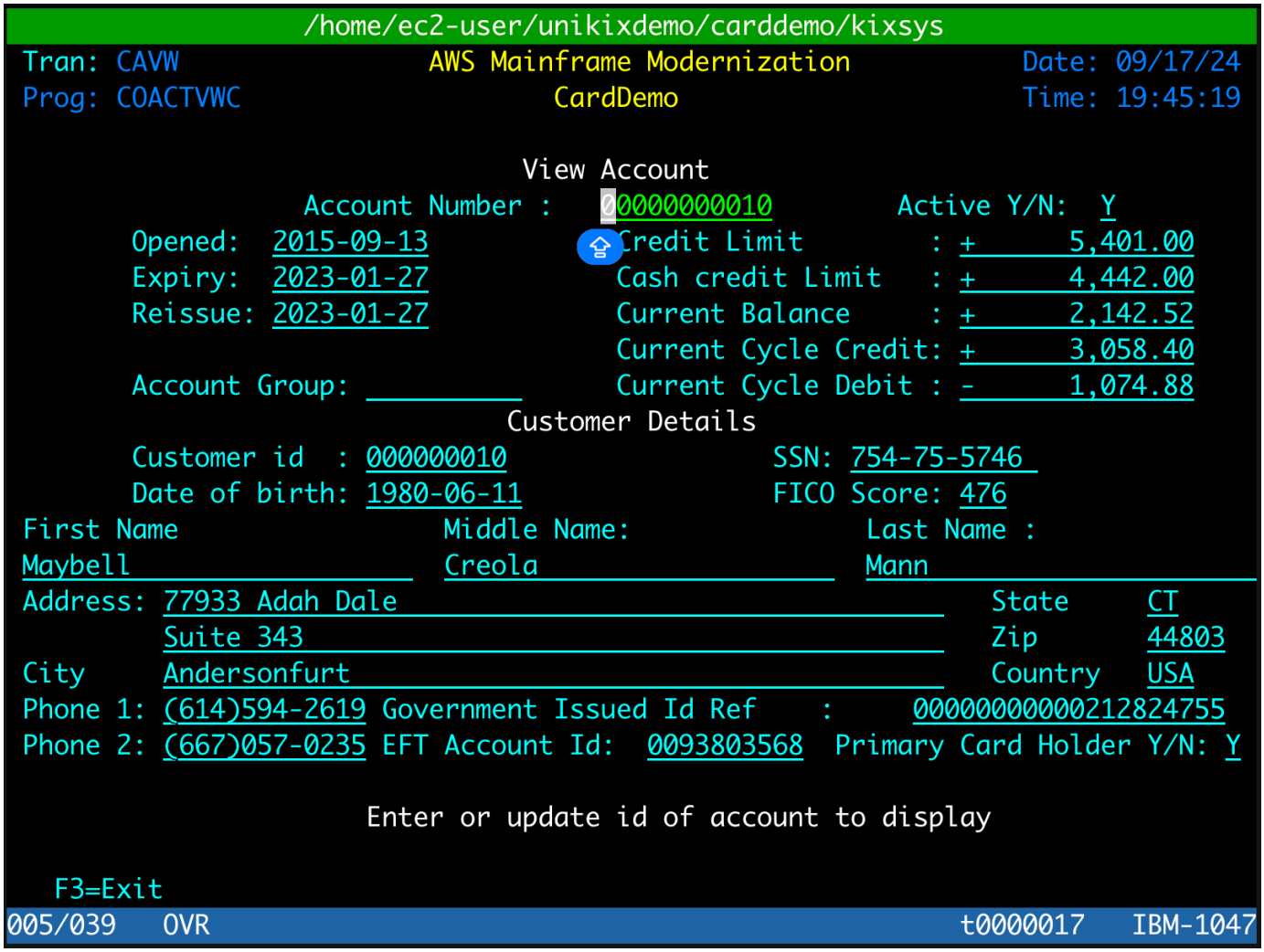
1. Nachdem Sie vom Terminalemulator TN327 0 aus eine Verbindung hergestellt haben, drücken Sie die Eingabetaste, um den TPE-Bildschirm zu löschen und die erste Transaktion einzugeben.
2. Geben Sie bei der ersten Transaktion CC00 (Anmeldebildschirm) den Benutzernamen und `USER001` `PASSWORD` das Passwort ein.




3. Wählen Sie im Hauptmenü die Option „01“, um Konten anzuzeigen.




4. Geben Sie im Bildschirm „Konto anzeigen“ eine Kontonummer ein (z. B. 00000000010). Sie sollten die Kontoinformationen sehen, die aus den migrierten Daten stammen.




5. Drücken Sie zweimal die Taste „PF03“, um zum Anmeldebildschirm zurückzukehren, und verlassen Sie das TN327 0-Terminal, indem Sie „Strg+C“ (Windows) oder „Cmd+C“ (Macbook) drücken.

Batch-Jobs

1. Navigieren Sie zum JCL-Verzeichnis.

```
cd $MBMSUB
```

2. Senden Sie den Job ab `MFCATGL1` und beobachten Sie die Ausgabe des Jobprotokolls.

```
BPESUB READCARD
```

3. Optional können Sie die Jobprotokolle aus dem `$SUBSYS_OUTDIR` Verzeichnis anzeigen.

```
ls -lrt $SUBSYS_OUTDIR/*
```

Sie haben die CardDemo Anwendung nun erfolgreich auf der NTT UniKix DATA-Laufzeit bereitgestellt und die laufende Anwendung verifiziert, indem Sie durch einige CICS-Onlinebildschirme und Batch-Jobs navigiert haben.

## Fehlerbehebung

Im Folgenden finden Sie einige häufig auftretende Fehler, die bei der Einrichtung der Anwendung auftreten können. CardDemo

### Fehler: Lizenzierungsfehler

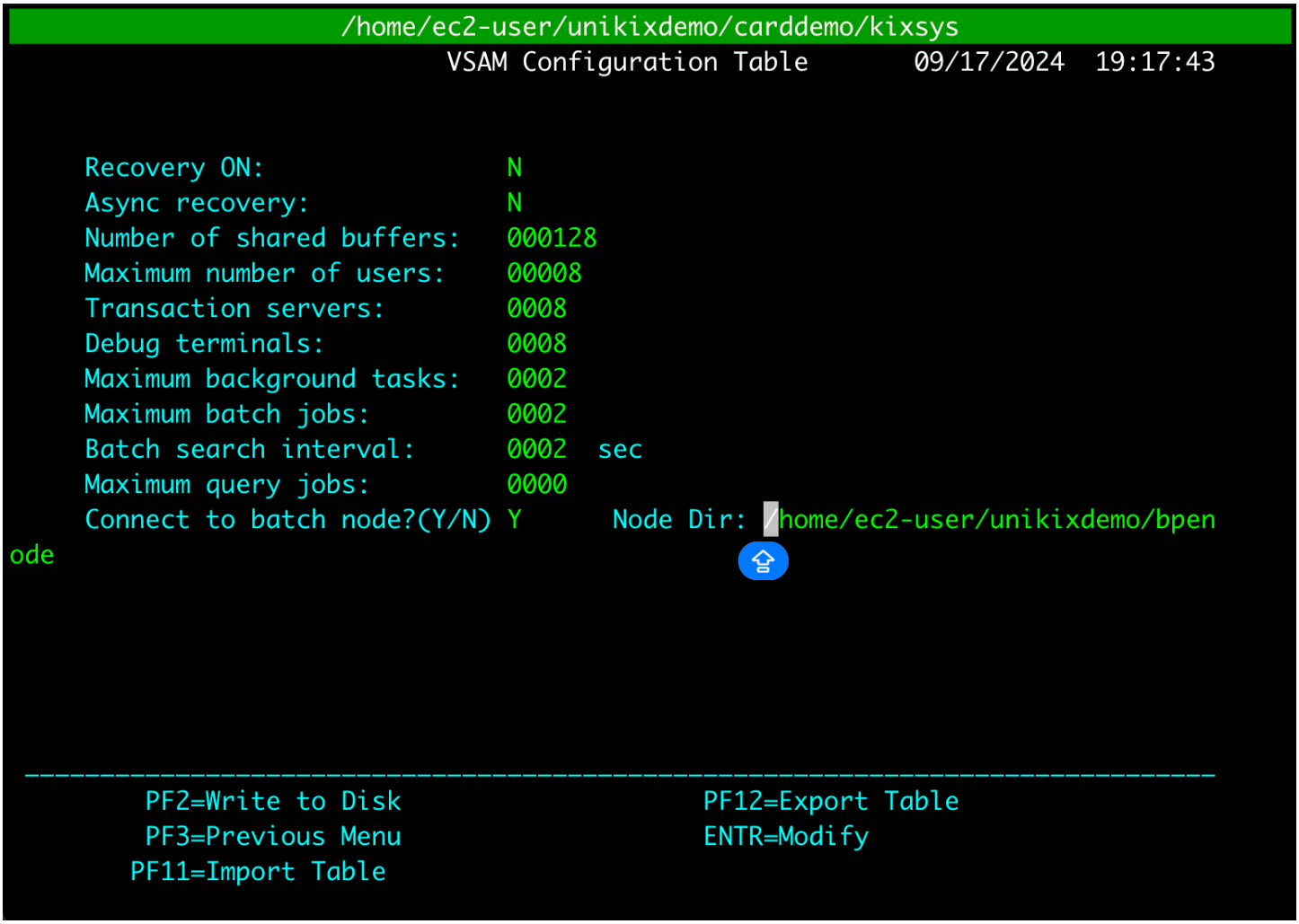
Wenn Sie während der Durchführung dieses Tutorials einen Lizenzfehler erhalten, kann es sein, dass der in Ihrem aktiviert IMDSv2ist EC2. Sie können dieses Problem lösen, indem Sie die Option für Instance-Metadaten IMDSv2auf Optional ändern, wie unter Restore the use of IMDSv1 im EC2 Amazon-Benutzerhandbuch beschrieben.

### Fehler: TPE ist nicht mit BPE verbunden

Wenn das TPE nicht mit BPE verbunden ist, stellen Sie sicher, dass die VSAM-Konfigurationstabelle korrekt mit dem BPE-Knotenverzeichnis konfiguriert ist. Um auf die VSAM-Konfigurationstabelle zuzugreifen, starten Sie den TN327 0-Terminal-Emulator mit dem folgenden Befehl:

```
kixterm
```

1. Geben Sie den Transaktionsnamen als ein. CTBL
2. Wählen Sie im Menü Table Manager die Option Standardtabellen.
3. Wählen Sie auf dem Bildschirm Standardtabellen die Option VSAM-Konfigurationstabelle.
4. Prüfen Sie, ob Connect zum Batch-Knoten herstellen? ist auf "Y“ gesetzt und das Knotenverzeichnis ist korrekt.


/home/ec2-user/unikixdemo/carddemo/kixsys
VSAM Configuration Table 09/17/2024 19:17:43
Recovery ON: N
Async recovery: N
Number of shared buffers: 000128
Maximum number of users: 00008
Transaction servers: 0008
Debug terminals: 0008
Maximum background tasks: 0002
Maximum batch jobs: 0002
Batch search interval: 0002 sec
Maximum query jobs: 0000
Connect to batch node?(Y/N) Y Node Dir: /home/ec2-user/unikixdemo/bpen
ode
PF2=Write to Disk
PF3=Previous Menu
PF11=Import Table
PF12=Export Table
ENTR=Modify

# Sicherheit bei der AWS Mainframe-Modernisierung

Cloud-Sicherheit AWS hat 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 so konfigurieren, dass Sie Ihre Sicherheits- und Compliance-Ziele 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
- Konformitätsvalidierung für die AWS Mainframe-Modernisierung
- Resilienz bei der AWS Mainframe-Modernisierung
- Sicherheit der Infrastruktur in AWS Mainframe Modernization
- Zugriff AWS Mainframe Modernization über einen AWS PrivateLink Schnittstellenendpunkt


# 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-Beitrag 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).
- Wird verwendet SSL/TLS , um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Richten Sie die API und die Protokollierung von Benutzeraktivitäten mit ein AWS CloudTrail. Informationen zur Verwendung von CloudTrail Pfaden zur Erfassung von AWS Aktivitäten finden Sie unter Arbeiten mit CloudTrail Pfaden im AWS CloudTrail Benutzerhandbuch.
- 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 140-3-validierte kryptografische Module benötigen, verwenden Sie einen FIPS-Endpunkt. Weitere Informationen über verfügbare FIPS-Endpunkte finden Sie unter Federal Information Processing Standard (FIPS) 140-3.

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 Anwendungen AWS-Services über die Konsole, API oder arbeiten. AWS CLI AWS SDKs 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 FSx Amazon-Dateisysteme an Ihre Laufzeitumgebung anzuhängen, werden die Daten in diesen Systemen gespeichert AWS. 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 Amazon EFS- und FSx Amazon-Verschlüsselungsoptionen 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 AWS Key Management Service , um transparente serverseitige Verschlüsselung (SSE) für alle abhängigen Ressourcen bereitzustellen, die Daten dauerhaft speichern, nämlich Amazon Simple Storage Service, Amazon DynamoDB und Amazon Elastic Block Store. 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, Anwendungen zu migrieren, die strikte Einhaltung von Verschlüsselungsvorschriften und gesetzlichen Anforderungen erfordern.

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-Volume zu verschlüsseln, das es erstellt und an Ihre AWS Mainframe Modernization EC2 Amazon-Instance 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:

- Erneutes Bereitstellen einer Anwendung.
- Ersetzen einer EC2 Amazon-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 |
|---|---|---|
| `Definition`<br><br>Enthält die Definition für eine bestimmte Anwendung. | Aktiviert | Aktiviert |
| `EnvironmentSummary`<br><br>Enthält Informationen über die Laufzeitumgebung. | Aktiviert | Aktiviert |
| `ApplicationSummary`<br><br>Enthält Informationen über die AWS Mainframe-Modernisierungsanwendung. | Aktiviert | Aktiviert |
| `DeploymentSummary`<br><br>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 CreateGrantAnfrage 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 DescribeKeyAnfragen 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 GenerateDataKeyAnfragen AWS KMS zur Verschlüsselung des Amazon EBS-Volumes, das EC2 Amazon-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

  - ReEncryptFrom
  - ReEncryptTo
  - GenerateDataKey
  - DescribeKey
  - CreateGrant
- Erstellen Sie einen Anwendungszuschuss oder stellen Sie ihn erneut bereit:
  - GenerateDataKey
- 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 Rocket Software (ehemals 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 FSx Amazon-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 erstellen, indem Sie den AWS Management Console oder den AWS KMS APIs verwenden.

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 Ruhestand 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 unterErstellen 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 unterErstellen Sie eine Anwendung AWS Mainframe Modernization.

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 nachzuverfolgen, 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 Beispielereignis zeichnet den Vorgang `DescribeKey` auf:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T19:40:26Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-12-06T20:23:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.251.233.182",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "keyId": "00dd0db0-0000-0000-ac00-b0c000SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
```

```
    },
    "sessionCredentialFromConsole": "true"
}
```

### CreateGrant

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 Verwendung zurückgezogen. Andere werden zurückgezogen, wenn Sie die Laufzeitumgebung löschen.

Das folgende Beispielereignis zeichnet den `CreateGrant` Vorgang für die Lambda-Ausführungsrolle auf, die dem Workflow Create Environment zugeordnet ist.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T20:11:45Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "m2.us-west-2.amazonaws.com"
    },
    "eventTime": "2022-12-06T20:23:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
```

```
    "awsRegion": "us-west-2",
    "sourceIPAddress": "m2.us-west-2.amazonaws.com",
    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "operations": [
            "Encrypt",
            "Decrypt",
            "ReEncryptFrom",
            "ReEncryptTo",
            "GenerateDataKey",
            "GenerateDataKey",
            "DescribeKey",
            "CreateGrant"
        ],
        "granteePrincipal": "m2.us-west-2.amazonaws.com",
        "retiringPrincipal": "m2.us-west-2.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Das folgende Beispielereignis 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": "AROA3YPCLM65MZFUPM4JO:EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",
        "arn": "arn:aws:sts::111122223333:assumed-role/EnvironmentWorkflow-
alpha-CreateEnvironmentLambdaS-1AU4A8VNQEEKN/EnvironmentWorkflow-alpha-
CreateEnvironmentLambda7-HfxDj5zz86tr",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:iam::111122223333:role/EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN",
                "accountId": "111122223333",
                "userName": "EnvironmentWorkflow-alpha-
CreateEnvironmentLambdaS-1AU4A8VNQEEKN"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T20:22:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-12-06T20:23:09Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "54.148.236.160",
    "userAgent": "aws-sdk-java/2.18.21 Linux/4.14.255-276-224.499.amzn2.x86_64
 OpenJDK_64-Bit_Server_VM/11.0.14.1+10-LTS Java/11.0.14.1 vendor/Amazon.com_Inc. md/
internal exec-env/AWS_Lambda_java11 io/sync http/Apache cfg/retry-mode/legacy",
    "requestParameters": {
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "operations": [
            "Encrypt",
            "Decrypt",
            "ReEncryptFrom",
            "ReEncryptTo",
            "GenerateDataKey",
            "GenerateDataKey",
            "DescribeKey",
            "CreateGrant"
        ],
        "granteePrincipal": "m2.us-west-2.amazonaws.com",
        "retiringPrincipal": "m2.us-west-2.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
}
```

### GenerateDataKey

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 Beispielereignis 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 Beispielereignis 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 Beispielereignis zeichnet den Vorgang `CreateGrant` auf:


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T21:51:45Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "m2.us-west-2.amazonaws.com"
    },
    "eventTime": "2022-12-06T22:47:04Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "m2.us-west-2.amazonaws.com",
    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:m2:app": "a1bc2defabc3defabc4defabcd"
            }
        },
        "retiringPrincipal": "m2.us-west-2.amazonaws.com",
```

```
        "operations": [
            "GenerateDataKey"
        ],
        "granteePrincipal": "m2.us-west-2.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

### GenerateDataKey

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 Beispielereignis zeichnet den Vorgang `GenerateDataKey` auf:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
```

```
        "principalId": "AROA3YPCLM65CLCEKKC7Z:ApplicationWorkflow-alpha-
CreateApplicationVersion-CstWZUn5R4u6",
        "arn": "arn:aws:sts::111122223333:assumed-role/ApplicationWorkflow-
alpha-CreateApplicationVersion-1IZRBZYDG20B/ApplicationWorkflow-alpha-
CreateApplicationVersion-CstWZUn5R4u6",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:iam::111122223333:role/ApplicationWorkflow-alpha-
CreateApplicationVersion-1IZRBZYDG20B",
                "accountId": "111122223333",
                "userName": "ApplicationWorkflow-alpha-
CreateApplicationVersion-1IZRBZYDG20B"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T23:28:32Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "m2.us-west-2.amazonaws.com"
    },
    "eventTime": "2022-12-06T23:29:08Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "m2.us-west-2.amazonaws.com",
    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:m2:app": "a1bc2defabc3defabc4defabcd",
            "aws:s3:arn": "arn:aws:s3:::supernova-processedtemplate-111122223333-us-
west-2/111122223333/a1bc2defabc3defabc4defabcd/1/cics-transaction/ZBNKE35.so"
        },
        "keySpec": "AES_256",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
```

```
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

## 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 Rolle „Amazon EC2 Scoped Down Instance“, sodass Amazon Anwendungsressourcen von Kunden von Amazon S3 herunterladen EC2 kann. Dieser Zuschuss wird zurückgezogen, wenn die Anwendung aus der Laufzeitumgebung gelöscht wird.

Das folgende Beispielereignis zeichnet den Vorgang `CreateGrant` auf:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
```

```
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T21:51:45Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "m2.us-west-2.amazonaws.com"
    },
    "eventTime": "2022-12-06T23:40:07Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "m2.us-west-2.amazonaws.com",
    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
        "operations": [
            "Decrypt"
        ],
        "constraints": {
            "encryptionContextSubset": {
                "aws:m2:app": "a1bc2defabc3defabc4defabcd"
            }
        },
        "granteePrincipal": "m2.us-west-2.amazonaws.com",
        "retiringPrincipal": "m2.us-west-2.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
    },
```

```
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Decrypt

Wenn Sie auf eine Bereitstellung zugreifen, EC2 ruft Amazon den `Decrypt` Vorgang auf, um den gespeicherten verschlüsselten Datenschlüssel zu verwenden, um verschlüsselte Kundendaten von Amazon S3 zu entschlüsseln und herunterzuladen.

Das folgende Beispielereignis zeichnet den Vorgang `Decrypt` auf:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROA3YPCLM65BSPZ37E6G:m2-hm-bqe367dxtfcpdbzmnhfzranisu",
        "arn": "arn:aws:sts::111122223333:assumed-role/
SupernovaEnvironmentInstanceScopeDownRole/m2-hm-bqe367dxtfcpdbzmnhfzranisu",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE3",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAIGDTESTANDEXAMPLE:Sampleuser01",
                "arn": "arn:aws:iam::111122223333:role/
SupernovaEnvironmentInstanceScopeDownRole",
                "accountId": "111122223333",
                "userName": "SupernovaEnvironmentInstanceScopeDownRole"
            },
```

```
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-12-06T23:19:29Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "m2.us-west-2.amazonaws.com"
    },
    "eventTime": "2022-12-06T23:40:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "m2.us-west-2.amazonaws.com",
    "userAgent": "m2.us-west-2.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "aws:m2:app": "a1bc2defabc3defabc4defabcdm",
            "aws:s3:arn": "arn:aws:s3:::supernova-processedtemplate-111122223333-us-
west-2/111122223333/a1bc2defabc3defabc4defabcdm/1/cics-transaction/BBANK40P.so"
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

## 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, erfolgt der Datenaustausch zwischen dem Terminalemulator und dem AWS Mainframe Modernization Service-Endpunkt für das TN327 0-Protokoll bei der Übertragung nicht verschlüsselt. Wenn für die Anwendung während der Übertragung eine Verschlüsselung erforderlich ist, sollten Sie möglicherweise einige zusätzliche Tunnelmechanismen implementieren.

AWS Die Mainframe-Modernisierung verwendet HTTPS, um den Dienst zu verschlüsseln. APIs 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 TN327 0-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 AWS kontrollieren. 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 AWS Mainframe Modernization


## 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. Anschließend müssen Sie Anforderungen an Ihren IAM-Administrator senden, um die Berechtigungen der Servicebenutzer 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 für die Selbstsignierung von Anforderungen finden Sie unter AWS Signature Version 4 für API-Anforderungen im IAM-Benutzerhandbuch.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen bereitstellen. 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 AWS Multi-Faktor-Authentifizierung (MFA) in IAM im IAM-Benutzerhandbuch.

## AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, 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. Verwenden Sie diese nur, 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 eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung 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 Anwendungsfälle für IAM-Benutzer im IAM-Benutzerhandbuch.

## IAM-Rollen

Eine IAM-Rolle ist eine Identität innerhalb von Ihnen AWS-Konto , die über bestimmte Berechtigungen verfügt. Sie ist einem IAM-Benutzer vergleichbar, jedoch nicht mit einer bestimmten Person verknüpft. Um vorübergehend eine IAM-Rolle in der zu übernehmen AWS Management Console, können Sie von einer Benutzer- zu einer IAM-Rolle (Konsole) 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 Methoden für die Übernahme einer Rolle 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 (Verbund) 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 Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.
- Serviceübergreifender Zugriff — Einige AWS-Services verwenden Funktionen in anderen AWS-Services. Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. 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, kombiniert 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 ausgeführte Anwendungen EC2 — Sie können eine IAM-Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API-Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein

Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAM-Benutzerhandbuch unter Verwenden einer IAM-Rolle, um Berechtigungen für Anwendungen zu gewähren, die auf EC2 Amazon-Instances ausgeführt werden.

## 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. Die Berechtigungen in den Richtlinien legen fest, ob eine 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 heißt, welcher Prinzipal Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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

Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 Inline-Richtlinie wählen, finden Sie unter Auswählen 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.

## Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON-Richtliniendokumentformat.

Amazon S3 und Amazon VPC sind Beispiele für Dienste, die Unterstützung ACLs bieten. AWS WAF Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

## 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.
- Dienststeuerungsrichtlinien (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 alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Root-Benutzer des AWS-Kontos Entitäten. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Ressourcenkontrollrichtlinien (RCPs) — RCPs sind JSON-Richtlinien, mit denen Sie die maximal verfügbaren Berechtigungen für Ressourcen in Ihren Konten festlegen können, ohne die IAM-Richtlinien aktualisieren zu müssen, die jeder Ressource zugeordnet sind, deren Eigentümer Sie sind. Das RCP schränkt die Berechtigungen für Ressourcen in Mitgliedskonten ein und kann sich auf die effektiven Berechtigungen für Identitäten auswirken, einschließlich der Root-Benutzer des AWS-Kontos, unabhängig davon, ob sie zu Ihrer Organisation gehören. Weitere Informationen zu Organizations RCPs, einschließlich einer Liste AWS-Services dieser Support-Leistungen RCPs, finden Sie unter Resource Control Policies (RCPs) 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 AWS Mainframe Modernization verwenden, sollten Sie sich darüber informieren, welche IAM-Funktionen im Rahmen der Mainframe-Modernisierung zur Verfügung stehen. AWS

IAM-Funktionen, die Sie mit der Mainframe-Modernisierung verwenden können AWS

| IAM-Feature | AWS Unterstützung für die Mainframe-Modernisierung |
|---|---|
| Identitätsbasierte Richtlinien | Ja |
| Ressourcenbasierte Richtlinien | Nein |
| Richtlinienaktionen | Ja |
| Richtlinienressourcen | Ja |
| Bedingungsschlüssel für die Richtlinie | Ja |
| ACLs | Nein |
| ABAC (Tags in Richtlinien) | Ja |
| Temporäre Anmeldeinformationen | Ja |
| Forward Access Sessions (FAS) | Ja |

| IAM-Feature | AWS Unterstützung für die Mainframe-Modernisierung |
| --- | --- |
| Servicerollen | Ja |
| Service-verknüpfte Rollen | 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 Definieren benutzerdefinierter IAM-Berechtigungen mit vom Kunden verwalteten 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 Kontoübergreifender Ressourcenzugriff in IAM 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 Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen kann.

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 heißt, welcher Prinzipal Aktionen für welche Ressourcen und 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 dieser Ressourcen die Ressource identifizieren, für die die IAM-Richtlinie gilt. ARNs Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) in der 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 deren Eigenschaften finden Sie unter Ressourcen ARNs, die durch die AWS Mainframe-Modernisierung definiert wurden in der Serviceautorisierungsreferenz. 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 zur Durchführung 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 |
|---|---|---|
| CancelBatchJobExecution | | Anwendung |
| CreateApplication | `iam:PassRole`<br>`kms:DescribeKey`<br>`kms:CreateGrant` | Anwendung |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
| --- | --- | --- |
| | `s3:GetObject`<br><br>`s3:ListBucket` | |
| CreateDataSetImportTask | `s3:GetObject` | Anwendung |
| CreateDataSetExportTask | `kms:DescribeKey`<br><br>`s3:PutObject` | Anwendung |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
| --- | --- | --- |
| CreateDeployment | elasticloadbalancing:AddTags<br>elasticloadbalancing:CreateListener<br>elasticloadbalancing:CreateTargetGroup<br>elasticloadbalancing:RegisterTargets<br>elasticloadbalancing:DeleteListener<br>elasticloadbalancing:DeleteTargetGroup<br>elasticloadbalancing:DeregisterTargets<br>elasticloadbalancing:DeleteLoadBalancer<br>logs:CreateLogDelivery<br>logs:GetLogDelivery<br>logs:UpdateLogDelivery<br>logs:DeleteLogDelivery<br>logs:ListLogDeliveries<br>logs:PutResourcePolicy | Anwendung |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
| --- | --- | --- |
| | `logs:DescribeResourcePolicies`<br><br>`logs:DescribeLogGroups` | |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
| --- | --- | --- |
| CreateEnvironment | ec2:CreateNetworkInterface<br><br>ec2:CreateNetworkInterfacePermission<br><br>ec2:DescribeNetworkInterfaces<br><br>ec2:DescribeSecurityGroups<br><br>ec2:DescribeSubnets<br><br>ec2:DescribeVpcAttribute<br><br>ec2:DescribeVpcs<br><br>ec2:ModifyNetworkInterfaceAttribute<br><br>elasticfilesystem:DescribeMountTargets<br><br>elasticloadbalancing:AddTags<br><br>elasticloadbalancing:CreateLoadBalancer<br><br>elasticloadbalancing:DeleteLoadBalancer<br><br>kms:DescribeKey<br><br>kms:CreateGrant<br><br>fsx:DescribeFileSystems | Umgebung |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
| --- | --- | --- |
| | `iam:CreateServiceLinkedRole` | |
| DeleteApplication | `elasticloadbalancing:DeleteListener`<br>`elasticloadbalancing:DeleteTargetGroup`<br>`logs:DeleteLogDelivery` | Anwendung |
| DeleteApplicationFromEnvironment | `elasticloadbalancing:DeleteListener`<br>`elasticloadbalancing:DeleteTargetGroup` | Anwendung<br>Umgebung |
| DeleteEnvironment | `elasticloadbalancing:DeleteLoadBalancer` | Umgebung |
| GetApplication | | Anwendung |
| GetApplicationVersion | | Anwendung |
| GetBatchJobExecution | | Anwendung |
| GetDataSetDetails | | Anwendung |
| GetDataSetImportTask | | Anwendung |
| GetDataSetExportTask | | Anwendung |
| GetDeployment | | Anwendung |
| GetEnvironment | | Umgebung |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
|---|---|---|
| ListApplications | | * |
| ListApplicationVersions | | * |
| ListBatchJobDefinitions | | * |
| ListBatchJobExecutions | | * |
| ListDataSetImportHistory | | * |
| ListDataSetExportHistory | | * |
| ListDataSets | | * |
| ListDeployments | | * |
| ListEngineVersions | | * |
| ListEnvironments | | * |
| ListTagsForResource | | * |
| StartApplication | | Anwendung |
| StartBatchJob | | Anwendung |
| StopApplication | | Anwendung |
| TagResource | | * |
| UntagResource | | * |

| AWS API-Operationen zur Mainframe-Modernisierung | Erforderliche Berechtigungen (API-Aktionen) | Ressourcen |
|---|---|---|
| UpdateApplication | `s3:GetObject`<br><br>`s3:ListBucket` | Anwendung |
| UpdateEnvironment | `kms:DescribeKey` | Umgebung |

## Schlüssel zu den Policy-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, AWS 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) in der Mainframe-Modernisierung AWS

Unterstützt ACLs: Nein

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLs ähneln ressourcenbasierten 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 Bedingungselement 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 Definieren von Berechtigungen mit ABAC-Autorisierung 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 von einer Benutzerrolle zu einer IAM-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/Auftragseingang oder Abschluss usw.).

> **Warning**
>
> Das Ändern der Berechtigungen für eine Servicerolle kann die Funktionalität der AWS Mainframe-Modernisierung beeinträchtigen. 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 ausgeführten Anwendungen annehmen EC2 können, können Sie diese Rolle wählen, wenn Sie eine Startvorlage oder eine 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 EC2 Amazon-Instances ausgeführt werden, im Amazon EC2 Auto Scaling 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 AWS Mainframe Modernization

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 (Konsole) im IAM-Benutzerhandbuch.

Einzelheiten zu den durch AWS Mainframe Modernization definierten Aktionen und Ressourcentypen, einschließlich des Formats der ARNs für die einzelnen Ressourcentypen, 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-Modernisierungskonsole 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 Richtlinienvalidierung mit 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 Sicherer API-Zugriff mit MFA 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-Modernisierungskonsole 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:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

# Fehlerbehebung bei Identität und Zugriff bei der AWS Mainframe-Modernisierung

Verwenden Sie die folgenden Informationen, um häufig auftretende 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. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, 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 Kontoübergreifender Ressourcenzugriff in IAM im IAM-Benutzerhandbuch.

## Verwenden von serviceverknüpften Rollen für AWS Mainframe Modernization

AWS Mainframe Modernization verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein einzigartiger Typ von IAM-Rolle, mit der direkt verknüpft ist. AWS Mainframe Modernization Mit Diensten verknüpfte Rollen sind vordefiniert AWS Mainframe Modernization und enthalten alle Berechtigungen, die der Dienst benötigt, um andere AWS Dienste in Ihrem Namen aufzurufen.

Eine dienstbezogene Rolle AWS Mainframe Modernization erleichtert die Einrichtung, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. AWS Mainframe Modernization definiert die Berechtigungen ihrer dienstbezogenen Rollen und AWS Mainframe Modernization kann, sofern nicht anders definiert, nur 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 AWS Mainframe Modernization Ressourcen geschützt, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entziehen können.

Informationen zu anderen Diensten, die dienstverknüpfte Rollen unterstützen, finden Sie unter AWS Dienste, die mit IAM funktionieren. Suchen Sie in der Spalte Dienstverknüpfte Rollen nach

den Diensten, für die Ja steht. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer serviceverknüpften Rolle für diesen Service anzuzeigen.

## Berechtigungen von serviceverknüpften Rollen für AWS Mainframe Modernization

AWS Mainframe Modernization verwendet die serviceverknüpfte Rolle mit dem Namen AWSServiceRoleForAWSM2— konfiguriert das Netzwerk so, dass es eine Verbindung zu Ihrer VPC herstellt und auf Ressourcen wie Dateisysteme zugreift.

Die AWSService RoleFor AWSM2 dienstverknüpfte Rolle vertraut darauf, dass die folgenden Dienste die Rolle übernehmen:

- `m2.amazonaws.com`

Die genannte Rollenberechtigungsrichtlinie AWSM2 ServicePolicy ermöglicht es AWS Mainframe Modernization , die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Erstelle, lösche, beschreibe und füge Berechtigungen an EC2 Amazon-Netzwerkschnittstellen für die AWS Mainframe Modernization Umgebung hinzu, um Konnektivität mit der Kunden-VPC herzustellen.
- Registrierung oder Deregistrierung von Einträgen bei Elastic Load Balancing — so stellen Kunden eine Verbindung zur AWS Mainframe Modernization Umgebung her.
- Beschreiben Sie das Amazon EFS- oder FSx Amazon-Dateisystem, falls verwendet.
- Senden Sie Metriken CloudWatch aus der Laufzeitumgebung an den Kunden.

```
{
 "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.

## Erstellen einer serviceverknüpften Rolle für AWS Mainframe Modernization

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, AWS Mainframe Modernization wird die serviceverknüpfte Rolle für Sie erstellt.

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, AWS Mainframe Modernization erstellt die dienstverknüpfte Rolle erneut für Sie.

## Bearbeiten einer serviceverknüpften Rolle für AWS Mainframe Modernization

AWS Mainframe Modernization erlaubt Ihnen nicht, die AWSService RoleFor AWSM2 dienstverknüpfte Rolle zu bearbeiten. Da möglicherweise verschiedene Entitäten auf die Rolle verweisen, kann der Rollenname 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 AWS Mainframe Modernization

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 AWS Mainframe Modernization Dienst 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 AWS Mainframe Modernization Ressourcen zu löschen, die verwendet werden von AWSService RoleFor AWSM2

- Löschen Sie die Laufzeitumgebungen in AWS 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 AWSService RoleFor AWSM2 serviceverknüpfte Rolle zu löschen. Weitere Informationen finden Sie unter Löschen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

### Unterstützte Regionen für serviceverknüpfte AWS Mainframe Modernization -Rollen

AWS Mainframe Modernization unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS -Regionen und -Endpunkte.

## Konformitätsvalidierung für die AWS 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 herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen 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, sodass Sie überprüfen können AWS , ob Sie die Sicherheitsstandards und Best Practices der Branche einhalten.

## 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 Dienst 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 für den Zugriff AWS Mainframe Modernization über das Netzwerk. 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 AWS PrivateLink Schnittstellenendpunkt

Sie können AWS PrivateLink damit eine private Verbindung zwischen Ihrer VPC und AWS Mainframe Modernization herstellen. Sie können darauf zugreifen, AWS Mainframe Modernization als ob es in Ihrer VPC wäre, ohne ein Internet-Gateway, ein NAT-Gerät, eine VPN-Verbindung oder AWS Direct Connect eine Verbindung zu verwenden. Instances in Ihrer VPC benötigen für den Zugriff AWS Mainframe Modernization keine öffentlichen IP-Adressen.

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. Hierbei handelt es sich um vom Anforderer verwaltete Netzwerkschnittstellen, die als Eingangspunkt für den Datenverkehr dienen, der für AWS Mainframe Modernization bestimmt ist.

Weitere Informationen finden Sie AWS PrivateLink im AWS PrivateLink Leitfaden unter Zugriff AWS-Services durch.

## Überlegungen zu AWS Mainframe Modernization

Bevor Sie einen Schnittstellen-Endpunkt für einrichten AWS Mainframe Modernization, lesen Sie die Überlegungen im AWS PrivateLink Handbuch.

AWS Mainframe Modernization unterstützt Aufrufe aller API-Aktionen über den Schnittstellenendpunkt.

## Erstellen Sie einen Schnittstellenendpunkt für AWS Mainframe Modernization

Sie können einen Schnittstellenendpunkt für die AWS Mainframe Modernization Verwendung entweder der Amazon VPC-Konsole oder der 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 AWS Mainframe Modernization Verwendung des folgenden Servicenamens:

```
com.amazonaws.region.m2
```

Wenn Sie privates DNS für den Schnittstellenendpunkt aktivieren, können Sie API-Anfragen an die AWS Mainframe Modernization Verwendung des standardmäßigen regionalen DNS-Namens stellen. Beispiel, `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 AWS Mainframe Modernization über den Schnittstellenendpunkt. Um den Zugriff zu kontrollieren, der AWS Mainframe Modernization von Ihrer VPC aus gewährt wird, 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 Aktionen AWS Mainframe Modernization

Im Folgenden finden Sie ein Beispiel für eine benutzerdefinierte Endpunktrichtlinie. Wenn Sie diese Richtlinie an Ihren Schnittstellen-Endpunkt anhängen, gewährt sie allen Prinzipalen auf allen Ressourcen den Zugriff auf die aufgeführten AWS Mainframe Modernization -Aktionen.

```
//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 Alarme 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 EC2 Amazon-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 EC2 Amazon-Instances und anderen Quellen überwachen CloudTrail, speichern und darauf zugreifen. CloudWatch Logs kann 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 Verlaufsinformationen zugreifen können und einen besseren Überblick darüber erhalten, wie Ihre Webanwendung oder der Service ausgeführt werden. Sie können auch Alarme 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 | Die CPU-Auslastung von Instanzen in der Umgebung.<br><br>Dimension: EnvironmentID<br><br>Einheiten: Prozent<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| InboundNetworkThroughput | Eingehender Netzwerkdurchsatz von Instanzen in der Umgebung.<br><br>Dimension: EnvironmentID<br><br>Einheiten: Byte pro Sekunde<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| MemoryUtilization | Die Speicherauslastung von Instanzen in der Umgebung.<br><br>Dimension: EnvironmentID<br><br>Einheiten: Prozent<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

| Metrik | Beschreibung |
| --- | --- |
| OutboundNetworkThroughput | Ausgehender Netzwerkdurchsatz der Instanzen in der Umgebung.<br><br>Dimension: EnvironmentID<br><br>Einheiten: Byte pro Sekunde<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

## Anwendungsmetriken

| Metrik | Beschreibung |
| --- | --- |
| BatchJobCompletedCount | Die Anzahl der abgeschlossenen Jobs während des Zeitintervalls.<br><br>Diese Metrik ist für Rocket Software (ehemals Micro Focus) und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Summe |
| BatchJobFailedCount | Die Anzahl der fehlgeschlagenen Jobs während des Zeitintervalls.<br><br>Diese Metrik ist für Rocket Software und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl |

| Metrik | Beschreibung |
| --- | --- |
| | Gültige Statistiken: Summe |
| JvmMemoryFree | Die Menge des verfügbaren Speichers, der derzeit nicht von der Java Virtual Machine verwendet wird.<br><br>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.<br><br>Dimension: applicationId<br><br>Einheiten: Byte<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| JvmMemoryMax | Die maximal zulässige Speichermenge für die Java Virtual Machine.<br><br>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.<br><br>Dimension: applicationId<br><br>Einheiten: Byte<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

| Metrik | Beschreibung |
| --- | --- |
| JvmMemoryUsed | Die Menge an Arbeitsspeicher, die von der Java Virtual Machine aktiv genutzt wird.<br><br>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.<br><br>Dimension: applicationId<br><br>Einheiten: Byte<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| ProcessesActiveCount | Die aktive Anzahl von Prozessen zur gleichzeitigen Ausführung von Diensten, die Anfragen verarbeiten.<br><br>Diese Metrik ist nur für die Runtime-Engine von Rocket Software verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Summe |

| Metrik | Beschreibung |
| --- | --- |
| SessionCount | Die Anzahl der HTTP-Sitzungen für die Anwendung.<br><br>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.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| SharedMemoryFree | Der Speicher, der dem Unternehmensserver zur Verfügung steht, um alle Informationen zu speichern, die er zur Ausführung von Transakti onen und Aufträgen benötigt.<br><br>Diese Metrik ist nur für die Rocket Software Runtime Engine verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Kilobyte<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

| Metrik | Beschreibung |
| --- | --- |
| SharedMemoryTotal | Die Gesamtmenge des gemeinsam genutzten Speichers, der dem Unternehmensserver zugewiesen wurde, um alle Informationen zu speichern, die er zur Ausführung von Transakti onen und Aufträgen benötigt.<br><br>Diese Metrik ist nur für die Rocket Software Runtime Engine verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Kilobyte<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |
| ThreadActiveCount | Die Anzahl der Engine-Threads, die Anfragen verarbeiten.<br><br>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.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

| Metrik | Beschreibung |
| --- | --- |
| TransactionCompletedCount | Die Anzahl der festgeschriebenen Transaktionen während des Zeitintervalls.<br><br>Diese Metrik ist für Rocket Software und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Summe |
| TransactionFailedCount | Die Anzahl der fehlgeschlagenen Transaktionen während des Zeitintervalls.<br><br>Diese Metrik ist für Rocket Software und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Anzahl<br><br>Gültige Statistiken: Summe |

| Metrik | Beschreibung |
| --- | --- |
| TransactionResponseTime | 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.<br><br>Diese Metrik ist für Rocket Software und für AWS Blu Age 3.7.0 und spätere Versionen verfügbar.<br><br>Dimension: applicationId<br><br>Einheiten: Millisekunden<br><br>Gültige Statistiken: Durchschnitt, Minimum, Maximum |

### 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 der Ereignisse 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 CloudTrail -Element userIdentity.

## Grundlegendes zu den Einträgen in der AWS Mainframe-Modernisierung in der

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": "AROAII6WZTHGYAEXAMPLE",
        "arn": "arn:aws:sts::444455556666:assumed-role/Admin/Mary_Major",
        "accountId": "444455556666",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAII6WZTHGYAEXAMPLE",
                "arn": "arn:aws:iam::444455556666:role/Admin",
```

```
                "accountId": "444455556666",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-06-01T20:38:22Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-06-01T20:40:39Z",
    "eventSource": "m2.amazonaws.com",
    "eventName": "CreateApplication",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.196.65",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:91.0) Gecko/20100101
 Firefox/91.0",
    "requestParameters": {
        "clientToken": "1abc23de-f45g-6789-h01i-jkl2m3456789",
        "name": "MyApp",
        "description": "",
        "engineType": "microfocus",
        "definition": {
            "content": "{}"
        },
        "tags": {}
    },
    "responseElements": {
        "applicationVersion": 1,
        "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
        "applicationArn": "arn:aws:m2:us-east-1:444455556666:app/
lsfhmwhw7fffrosff2lncwqcua",
        "applicationId": "lsfhmwhw7fffrosff2lncwqcua"
    },
    "requestID": "36982d38-fcde-4bfe-a89a-7bd78d43c926",
    "eventID": "d7f0fc36-46ae-4157-9a79-c79f385fda98",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "444455556666",
    "eventCategory": "Management"
}
```

# Problembehandlung bei der AWS Mainframe-Modernisierung

Mithilfe der Informationen in diesem Abschnitt können Sie häufig auftretende Fehler in Anwendungen und Laufzeitumgebungen für die AWS Mainframe-Modernisierung mithilfe der AWS Blu Age- und Rocket Software-Engines beheben.

Themen


- Fehler zur Fehlerbehebung: Beim Warten auf die Entsperrung des Datensatznamens wurde eine Zeitüberschreitung überschritten
- Fehler zur Fehlerbehebung: Auf eine Anwendungs-URL kann nicht zugegriffen werden
- Fehlerbehebung: AWS Blu Insights lässt sich nicht von der Konsole aus öffnen
- Fehler zur Fehlerbehebung: Die Umgebung ist fehlerhaft
- Behebung von Lizenzproblemen für Rocket Software (ehemals Micro Focus)


## Fehler zur Fehlerbehebung: Beim Warten auf die Entsperrung des Datensatznamens wurde eine Zeitüberschreitung überschritten

Auf dieser Seite wird beschrieben, wie Sie Ihren Fehler beheben können, wenn Sie feststellen, dass eine andere Anwendung in einer Umgebung einen gemeinsam genutzten Datensatz sperrt.

- 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.

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 den Vorgang nicht fortsetzen und zeigt den `Failed` Status an.

## Häufige Ursache

Die Anwendung `example-app-1` versucht, einen Datensatz `example-record-1` für einen Schreibvorgang zu sperren. Durch diesen Vorgang wird sowohl eine Sperre für den Datensatz`example-dataset-1`, der 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`. Der Datensatz und der Datensatz sind bereits gesperrt, `example-app-2` wartet also darauf, dass 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 wird und eine Timeout-Ausnahme ausgelöst wird. Diese Deadlock-Situation verhindert, dass alle Anwendungen die Verbindung erreichen. `example-dataset-1`

## Auflösung

Um die Situation sofort zu lösen, können Sie das Aufheben der Sperre 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 (Redis OSS), um gemeinsame 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 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 zu Ihrem 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. |
| SCHLÜSSEL * *YOUR_DATASET_NAME* | Holen Sie sich einen Datensatz-Sperrschlüssel. |
| DEL *THE_RETURNED_KEY* | Löscht eine Datensatzsperre. |
| FLUSHDB | Säubere das gesamte Redis.<br><br>**Warning**<br>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` |

## 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. Deadlocks können 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 unterAWS 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 Datensatzsperre aufgehoben, und andere Datensätze aus demselben Datensatz 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.

# Fehler zur Fehlerbehebung: Auf eine Anwendungs-URL kann nicht zugegriffen werden

Auf dieser Seite wird beschrieben, wie Sie den Fehler beheben können, wenn Sie nicht auf die URL einer laufenden AWS Mainframe-Modernisierungsanwendung zugreifen können.

- Motor: AWS Blu Age und Rocket Software (ehemals 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.

## Häufige Ursache

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.

## Auflösung

Prüfen Sie, ob die Amazon VPC-Sicherheitsgruppe, die der Laufzeitumgebung zugeordnet ist, Datenverkehr zur Umgebung über die entsprechenden Anwendungsports zulässt. Gehen Sie wie folgt vor, um die Sicherheitsgruppenregeln zu überprüfen:

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole 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 oder Anywhere IPV4 zu wählen. IPV6 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.

# Fehlerbehebung: AWS Blu Insights lässt sich nicht von der Konsole aus öffnen

Auf dieser Seite wird beschrieben, wie Sie das Problem lösen können, dass die Blu Insights-Seite nicht von der AWS Mainframe Modernization Console aus geöffnet wird.

- 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.

## Häufige Ursache

Die Rolle, die Sie für den Zugriff auf Blu Insights verwenden, verfügt nicht über ausreichende Berechtigungen.

## Auflösung

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": "*"
            }
    ]
}
```

Stellen Sie sicher, dass Sie `region` und `account` durch das richtige AWS-Region und ersetzen AWS-Konto.

# Fehler zur Fehlerbehebung: Die Umgebung ist fehlerhaft

Auf dieser Seite wird beschrieben, wie Sie den Fehler beheben können, wenn Sie eine Benachrichtigung erhalten, dass eine Ihrer AWS Mainframe-Modernisierungsumgebungen fehlerhaft ist.

- Motor: AWS Blu Age und Rocket Software (ehemals Micro Focus)

- Komponente: Umgebungen

Wenn Sie eine Benachrichtigung erhalten, die besagt, dass eine Ihrer AWS Mainframe-Modernisierungsumgebungen fehlerhaft geworden ist, gilt dies auch für Sie. Sie werden über eine der folgenden Quellen benachrichtigt:

- Der Status der fehlerhaften Umgebung wird in Ihrer AWS Mainframe-Modernisierungskonsole angezeigt.
- E-Mail-Benachrichtigung über den Status der fehlerhaften Umgebung von. AWS Health
- Ein verwandtes Ereignis von AWS Mainframe Modernization wird in Ihrem AWS Health Dashboard unter Ihr Kontostatus angezeigt.

## Häufige Ursache

Der Fehler tritt auf, wenn auf die Ressourcen in Ihrem AWS Konto, die der AWS Mainframe-Modernisierungsumgebung zugeordnet sind, nicht zugegriffen werden kann. Ein häufiger Grund für dieses Problem ist, dass die Ressourcen, die sich auf die Umgebung beziehen, geändert oder gelöscht werden.

## Auflösung

Spezifische Hinweise erhalten Sie anhand des Fehlercodes, den Sie in der E-Mail von AWS Health oder über Ihre AWS Mainframe Modernization Console erhalten haben.

Fehlercode:

- Speicher nicht erreichbar

Dieser Fehler weist darauf hin, dass der angehängte Speicher (Amazon Elastic FSx File System oder Amazon-Dateisysteme) für die Umgebung nicht korrekt bereitgestellt werden konnte. Gehen Sie wie folgt vor, um Details zu einer fehlerhaften Umgebung zu überprüfen:

1. Öffnen Sie die AWS Mainframe-Modernisierungskonsole unter. https://console.aws.amazon.com/m2/
2. Wählen Sie die fehlerhafte Umgebung und dann Konfiguration aus.
3. Wählen Sie Attached Storage, um die mit dieser Umgebung verknüpften Speicherressourcen anzuzeigen.

4. Überprüfen Sie die netzwerkbezogenen Konfigurationen, z. B. die Sicherheitsgruppe, das Subnetz und die Amazon VPC, die dem Speicher zugeordnet sind. Wenn diese Konfigurationen falsch sind, versuchen Sie, sie wiederherzustellen, um das Problem zu lösen.

> **Note**
>
> Wenn der Speicher gelöscht wurde, kann die Umgebung nicht wiederhergestellt werden. In diesem Fall sollten Sie erwägen, die fehlerhafte Umgebung zu löschen.

# Behebung von Lizenzproblemen für Rocket Software (ehemals Micro Focus)

Auf dieser Seite wird beschrieben, wie Sie Lizenzprobleme mit der Rocket Software Runtime Engine lösen können

- Motor: Rocket Software
- Komponente: Amazon EC2

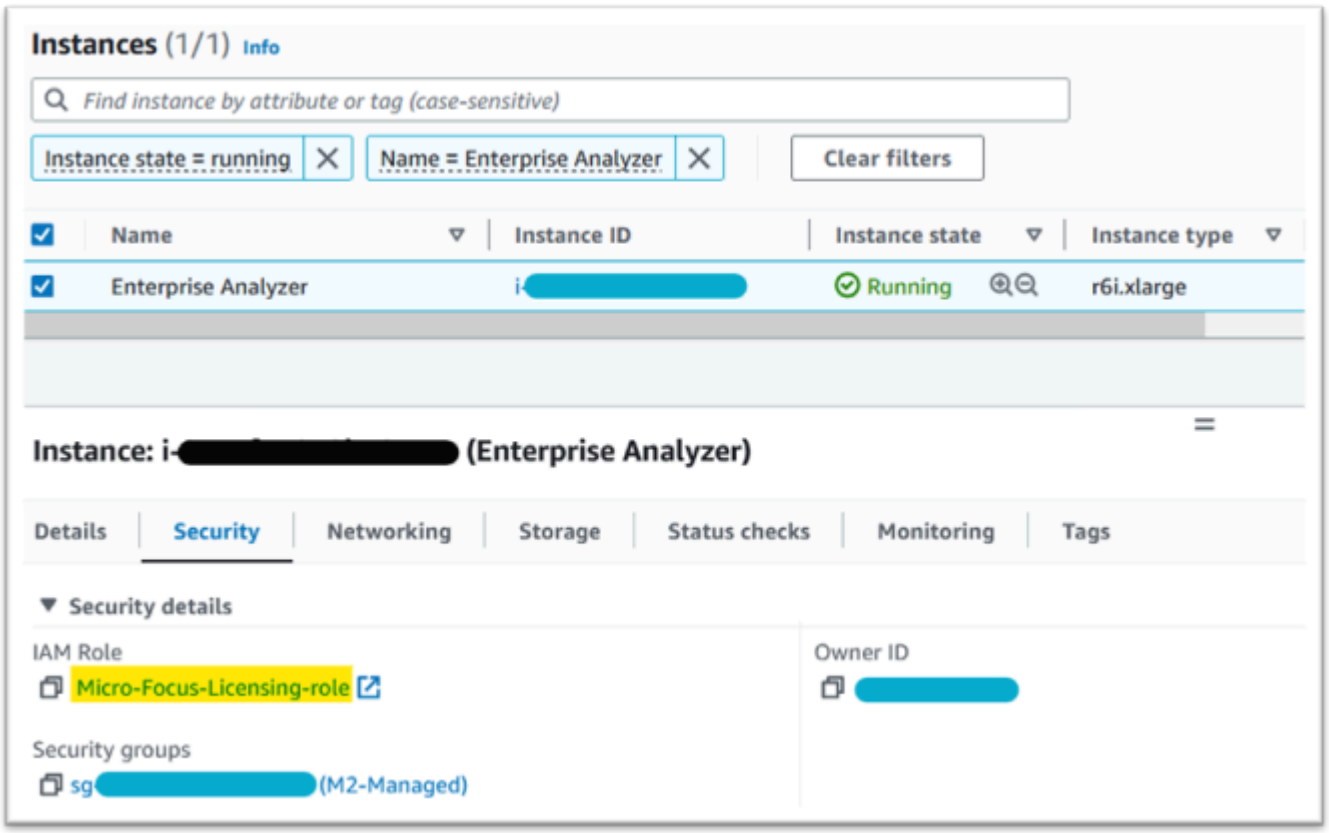
Wenn Sie Probleme beim Zugriff auf oder bei der Verwendung von haben AMIs, können Ihnen die folgenden Informationen weiterhelfen.

Themen


- Stellen Sie sicher, dass die EC2 Amazon-Instance die IAM-Lizenzierungsrolle hat
- Verwenden Sie den Erreichbarkeitsanalysator
- Führen Sie den License-Daemon aus
- Lizenzprobleme mit Enterprise Server oder Enterprise Build Tools unter Linux nach dem Patchen des Betriebssystems


## Stellen Sie sicher, dass die EC2 Amazon-Instance die IAM-Lizenzierungsrolle hat

Dies kann auf der Registerkarte Sicherheit in den EC2 Amazon-Instanzdetails überprüft werden. Dies kann mithilfe der Sicherheitsoption im Drop-down-Menü „Aktionen“ geändert werden.




## Verwenden Sie den Erreichbarkeitsanalysator

Suchen Sie den Reachability Analyzer auf der AWS Network Manager Konsolenseite.

Erstellen und analysieren Sie einen Pfad zwischen der vom AMI erstellten EC2 Amazon-Instance und dem Amazon S3-VPC-Endpunkt.

Wenn die EC2 Amazon-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. Ignoriere die SLF4 J-Meldungen und suche 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. Ignoriere die SLF4 J-Meldungen und suche nach der ersten Ausnahme.

Unter Linux starte:

```
java -jar /var/microfocuslicensing/bin/aws-license-daemon.jar
```

Ignoriere die SLF4 J-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.

## Lizenzprobleme mit Enterprise Server oder Enterprise Build Tools unter Linux nach dem Patchen des Betriebssystems

Wenn Sie nach dem Patchen des Betriebssystems Lizenzprobleme mit Enterprise Server oder Enterprise Build Tools unter Linux haben, aktualisieren Sie den Lizenz-Daemon, indem Sie ein Patch-Skript herunterladen und ausführen. Verwenden Sie dazu die folgenden Befehle in der Befehlszeile:

```
sudo curl https://d148y999krizvm.cloudfront.net/patch/v8/linux/patch.sh -o /var/
microfocuslicensing/bin/patch.sh
```

```
sudo chmod +x /var/microfocuslicensing/bin/patch.sh
sudo /var/microfocuslicensing/bin/patch.sh
sudo ./startmfcesd.sh
```

> **Note**
>
> Dieses Patch-Skript funktioniert auch mit Version 9, auch wenn der Download-Pfad für Version 8 gilt.

# Dokumentverlauf für das AWS Mainframe Modernization User Guide

In der folgenden Tabelle werden die Dokumentationsversionen für AWS Mainframe Modernization beschrieben.

| Änderung | Beschreibung | Datum |
| --- | --- | --- |
| AWS Blu Age-Versionshinweise 4.7.0 und 4.8.0 | Die Version 4.7.0 von AWS Blu Age Runtime und AWS Blu Age Transformation Engines beinhaltet grundlegende Abhängigkeitsupgrades auf Spring Boot 3.4.2 und Angular 19, wodurch Sicherheit und Leistung verbessert und gleichzeitig das Frontend-Anwendungsframework modernisiert wird. Die Version 4.8.0 enthält wichtige Funktionen wie die Unterstützung der IBM IMS-Datenbanktransformation und eine neue In-Memory-Cache-Funktion, die die Leistung datenintensiver Programme zur 00-Modernisierung verbessert. AS4 | 18. Mai 2025 |
| AWS Versionshinweise zu Blu Age 4.6.0 | Diese Version von AWS Blu Age Runtime und AWS Blu Age Transformation Engine konzentriert sich auf neue Funktionen und Verbesserungen für zOS und AS4 00. | 24. Januar 2025 |

| | | |
|---|---|---|
| AWS Versionshinweise zu Blu Age 4.5.0 | Diese Version von AWS Blu Age Runtime und AWS Blu Age Transformation Engine konzentriert sich auf die wichtigsten Funktionen der JCL-Unterstützung, die Unterstützung von Bindungsverzeichnissen und Aktivierungsgruppen für modernisierte AS/400-Anwendungen sowie aktualisierte Abhängigkeiten. | 20. Dezember 2024 |
| AWS Blu Age Häufig gestellte Fragen | In dieser umfassenden Liste von erfahren Sie mehr über die Refactoring-Funktion von AWS Blu Age. FAQs | 20. Dezember 2024 |
| Amazon Q Developer-Transformation für Mainframe | Erfahren Sie mehr über die Transform for Mainframe -Funktion von Amazon Q Developer, mit der Sie Ihre älteren COBOL-Mainframe-Anwendungen auf Java-Anwendungen umstellen können. | 2. Dezember 2024 |
| AWS Versionshinweise zu Blu Age 4.4.0 | Diese Version der AWS Blu Age Runtime and Transformation Engines konzentriert sich auf die Aktualisierung kritischer Abhängigkeiten und unterstützter Technologien bei gleichzeitiger Steigerung der Leistung mehrerer Funktionen. | 13. November 2024 |

| | | |
|---|---|---|
| Stellen Sie AWS Blu Age Runtime auf Containern bereit | Sie erfahren, wie Sie AWS Blu Age Runtime auf Containern einrichten, um sie auf Amazon ECS (verwaltet von Amazon EC2 oder AWS Fargate) und Amazon EKS, verwaltet von Amazon, bereitzustellen EC2. | 28. Oktober 2024 |
| Die LDAP-Benutzeranmeldeinformationen wurden aktualisiert | Sie können jetzt LDAP-Benutzeranmeldeinformationen für die Authentifizierung und Autorisierung entweder über die AWS Konsole oder AWS CLI (oder das SDK) erstellen und verwalten. | 21. Oktober 2024 |
| Konfigurieren Sie die von Rocket Software verwaltete Anwendung | Sie können Ihre Anwendungen jetzt mit der Rocket Software Runtime Engine konfigurieren, um zusätzliche Eigenschaften, einschließlich Integrationen, anzupassen. | 21. Oktober 2024 |
| Replatforming mit NTT DATA Unikix Runtime | Erfahren Sie, wie Sie mithilfe von Amazon Machine Images (AMIs) mithilfe von NTT DATA eine maßgeschneiderte Umgebung für das Rehosting und die Neuplattformierung AWS von Mainframe-Anwendungen erstellen können. | 25. September 2024 |

| | | |
|---|---|---|
| AWS Mainframe-Modernisierung, Anwendungstests (IAM) | Sie können sich mit den verfügbaren IAM-Funktionen und -Richtlinien über die Verwaltung des Zugriffs für AWS Mainframe Modernization Application Testing informieren. | 20. September 2024 |
| AWS Veröffentlichungen von Blu Age | In diesem Abschnitt des Benutzerhandbuchs zur AWS Mainframe-Modernisierung finden Sie alle Informationen zur AWS Blu Age-Versionierung, die Versionshinweise zu AWS Blu Age, Anweisungen zum Upgrade von AWS Blu Age und den gesamten Lebenszyklus von AWS Blu Age. | 16. September 2024 |
| AWS Lebenszyklus der Komponenten für die Mainframe-Modernisierung | Auf dieser Seite wird der Lebenszyklus der einzelnen Komponenten der AWS Mainframe-Modernisierung erfasst, einschließlich ihrer Versionsupgrades, des allgemeinen Release-Plans und der Pläne zum Ende des Supports und der Altersvorsorge. | 5. September 2024 |

| | | |
|---|---|---|
| Assembler-Konvertierung mit mLogica | AWS Mainframe-Modernisierung Die Codekonvertierung mit mLogica ist eine AWS Mainframe-Modernisierungsfunktion, die z/OS-Mainframe-Assembler-Code automatisch in COBOL konvertiert. | 22. Juli 2024 |
| Veröffentlichung von Application Testing (GA) | Dokumente zur allgemeinen Verfügbarkeit für Anwendungstests. AWS Mainframe Modernization Application Testing bietet automatisierte funktionale Äquivalenztests für Ihre Migrationsprojekte . Diese Version enthält seit der Vorschauversion eine Datenschutzseite, Konsolen-Workflows und Aktualisierungen anderer Dokumenteiten. | 12. Juni 2024 |
| Aktualisiertes Tutorial zu Managed Runtime für Rocket Software | Dieses Tutorial zeigt, wie die CardDemo Beispielanwendung in einer von AWS Mainframe Modernization verwalteten Laufzeitumgebung mit der Rocket Software Runtime Engine bereitgestellt und ausgeführt wird. | 5. Februar 2024 |

| | | |
|---|---|---|
| Versionshinweise für AWS Blu Age Runtime and Moderniza tion Tools Version 3.9.0. | Diese Version von AWS Blu Age Runtime and Moderniza tion Tools konzentriert sich auf mehrere bereichsübergreife nde Verbesserungen im gesamten Produkt, mit denen die Leistung in Hochverfü gbarkeitsarchitekturen gesteigert werden soll, sowie auf neue Funktionen, um die Ausführung von Jobs auf die nächste Stufe zu heben. | 18. Dezember 2023 |
| Übertragen Sie Dateien zwischen Mainframe und AWS | Neue Funktion zum Übertrage n von Dateien vom Quell-Mai nframe auf. AWS | 8. November 2023 |
| Transaktionen für Anwendung en verwalten | Neue Funktion zum Anzeigen und Bearbeiten von Transakti onen für Anwendungen zur AWS Mainframe-Modernis ierung veröffentlicht. | 16. Oktober 2023 |

| | | |
|---|---|---|
| Versionshinweise für AWS Blu Age Runtime and Moderniza tion Tools Version 3.6.0. | Diese Version von AWS Blu Age Runtime and Moderniza tion Tools bietet neue Funktionen für ältere Migration en von zOS und AS4 00. Diese sind hauptsächlich darauf ausgerichtet, die CICS-Unterstützungsmechanis men zu erweitern, die JCL-Funktionen zu ergänzen, die Leistung bei gleichzeitigen und umfangreichen Funktionen zu optimieren und Funktione n 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-P aket für eine einfachere EC2 Bereitstellung auf Amazon | AWS Mainframe-Modernis ierung mit AWS Blu Age Runtime ist jetzt mit mehr Flexibilität für die Konfigura tion des gesamten Stacks und die Bereitstellung auf EC2 Amazon-Instances in Ihrem AWS-Konto verfügbar. | 6. Juli 2023 |

| | | |
|---|---|---|
| Einmaliges Anmelden bei AWS Blu Age Blu Insights. | AWS Blu Age Blu Insights ist ab dem Single AWS Management Console Sign-On verfügbar. | 31. März 2023 |
| GA-Veröffentlichung | GA-Version des AWS Mainframe Modernization User Guide. | 8. Juni 2022 |
| Erstversion | 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.